

# ColdFusion Developer's Journal

ColdFusionJournal.com

September 2000 Volume: 2 Issue: 9

**XML DevCon FALL 2000** November 12-15, 2000  
**Announcing... Wireless DevCon** December 3-5, 2000

## Editorial

**Up with Allaire, Intel and CFDJList**  
 Robert Diamond page 5

## Foundations

**If You Meet Buddha on the Road, Kill Him**  
 Hal Helms page 14

## <BF> ON <CF>

**Just Your Type**  
 Ben Forta page 18

## Product Reviews

**eWebEditPro 1.8.0.2 from Ektron**  
 Michael Smith page 22

**HotQuery from Array Software**  
 Jerry Bradenbaugh page 56

## CF Techniques

**Permission Framework**  
 Kelly Brown page 26



**Verity's Search Technology:**  
 seamless to implement

page 6

Get a **FREE** Pass  
 INTERNATIONAL CONFERENCE  
 FOR JAVA TECHNOLOGY  
**DON'T MISS IT!**  
 The largest gathering of Java developers this fall!  
 Sept. 24-27

**CFDJ Feature: Unlocking Verity's Potential** 6  
*Searching and indexing technology for your Web site* David C. Smith

**CFDJ Feature: Got a Problem with Duplicate Records?** 28  
*Maybe it's the multithreading in the CF server* Raymond H. Thompson

**Get a Grip on the Clickstream** 34  
*Providing a brave new world of opportunity* Lee Surma

**The Question: What's FTP?** 36  
*That's the response I got from most of our clients...* Kevin Schmidt

**Customize ColdFusion Authentication** 40  
*Use this methodology in your ColdFusion application* Sarge

**CFDJ Feature: Another Custom Tag DHTML Wrapper** 46  
*An inside look* Christian Schneider

**Facing the Challenges Part 2** 60  
*Some of the problems, with their solutions, of venturing into wireless programming with ColdFusion* Charles Arehart

# **AbleCommerce**

**[www.ablecommers.com](http://www.ablecommers.com)**

# **Fig Leaf Software**

**[www.figleaf.com](http://www.figleaf.com)**

# **AbleCommerce**

**[www.ablecommers.com](http://www.ablecommers.com)**

STEVEN D. DRUCKER, JIM ESTEN, BEN FORTA,  
STEVE NELSON, RICHARD SCHULZE, PAUL UNDERWOOD

**EDITOR-IN-CHIEF** ROBERT DIAMOND  
**ART DIRECTOR** JIM MORGAN  
**EXECUTIVE EDITOR** M'LOU PINKHAM  
**MANAGING EDITOR** CHERYL VAN SISE  
**ASSOCIATE EDITOR** NANCY VALENTINE  
**PRODUCT REVIEW EDITOR** TOM TAUILLI  
**TIPS & TECHNIQUES EDITOR** MATT NEWBERRY

### WRITERS IN THIS ISSUE

CHARLES AREHART, JERRY BRADENBAUGH, KELLY BROWN,  
ROBERT DIAMOND, BEN FORTA, HAL HELMS, SARGE,  
KEVIN SCHMIDT, CHRISTIAN SCHNEIDER, DAVID C. SMITH,  
MICHAEL SMITH, LEE SURMA, RAY THOMPSON

### SUBSCRIPTIONS

SUBSCRIBE@SYS-CON.COM

FOR SUBSCRIPTIONS AND REQUESTS FOR BULK ORDERS,  
PLEASE SEND YOUR LETTERS TO  
SUBSCRIPTION DEPARTMENT.

SUBSCRIPTION HOTLINE 800 513-7111

COVER PRICE \$8.99/ISSUE

DOMESTIC \$79/YR. (12 ISSUES)

CANADA/MEXICO \$99/YR

OVERSEAS \$129/YR

BACK ISSUES \$12 EACH

**PRESIDENT AND CEO** FUAT A. KIRCAALI  
**VICE PRESIDENT, PRODUCTION** JIM MORGAN  
**VICE PRESIDENT, MARKETING** CARMEN GONZALEZ  
**GROUP PUBLISHER** LISE ST. AMANT  
**COMPTROLLER** BRUCE MILLER  
**ADVERTISING ACCOUNT MANAGER** ROBYN FORMA  
**ADVERTISING ACCOUNT MANAGER** MEGAN RING  
**ADVERTISING ASSISTANT** CHRISTINE RUSSELL  
**ADVERTISING INTERN** ALISON NOVICK  
**GRAPHIC DESIGNER** ALEX BOTERO  
**GRAPHIC DESIGNER** ABRAHAM ADDO  
**GRAPHIC DESIGNER** CATHRYN BURAK  
**GRAPHIC DESIGN INTERN** LOUIS F. CUFFARI  
**GRAPHIC DESIGN INTERN** AARATHI VENKATARAMAN  
**WEBMASTER** BRUNO Y. DECAUDIN  
**WEB DESIGNER** STEPHEN KILMURRAY  
**WEB DESIGNER** GINA ALAYAN  
**CUSTOMER SERVICE** ELLEN MOSKOWITZ  
**SYS-CON EVENTS MANAGER** ANTHONY D. SPITZER  
**JDJ STORE.COM** AMANDA MOSKOWITZ

### EDITORIAL OFFICES

SYS-CON MEDIA, INC. 135 CHESTNUT RIDGE RD.,  
MONTVALE, NJ 07645  
TELEPHONE: 201 802-3000 FAX: 201 782-9600

COLD FUSION DEVELOPER'S JOURNAL (ISSN #1523-9101)  
IS PUBLISHED MONTHLY (12 TIMES A YEAR)  
FOR \$79 BY SYS-CON PUBLICATIONS, INC.,  
135 CHESTNUT RIDGE RD., MONTVALE, NJ 07645

### POSTMASTER

SEND ADDRESS CHANGES TO:  
COLD FUSION DEVELOPER'S JOURNAL  
SYS-CON MEDIA, INC.

135 CHESTNUT RIDGE RD., MONTVALE, NJ 07645

### © COPYRIGHT

COPYRIGHT © 2000 BY SYS-CON MEDIA, INC.

ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE  
REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS,  
ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY OR ANY  
INFORMATION STORAGE AND RETRIEVAL SYSTEM,  
WITHOUT WRITTEN PERMISSION.

FOR PROMOTIONAL REPRINTS, CONTACT REPRINT COORDINATOR.

SYS-CON PUBLICATIONS, INC., RESERVES THE RIGHT TO REVISE,  
RE-PUBLISH AND AUTHORIZE ITS READERS TO USE  
THE ARTICLES SUBMITTED FOR PUBLICATION.

### WORLDWIDE DISTRIBUTION

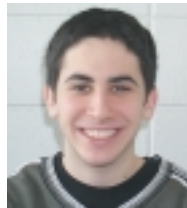
BY CURTIS CIRCULATION COMPANY 730 RIVER ROAD,  
NEW MILFORD, NJ 07646-3048 PHONE: 201 634-7400

### DISTRIBUTED IN USA

BY INTERNATIONAL PERIODICAL DISTRIBUTORS  
674 VIA DE LA VALLE, SUITE 204, SOLANA BEACH, CA 92075  
619 481-5928

ALL BRAND AND PRODUCT NAMES USED ON THESE PAGES  
ARE TRADE NAMES, SERVICE MARKS OR TRADEMARKS  
OF THEIR RESPECTIVE COMPANIES.

# Up with Allaire, Intel and CFDJList



BY ROBERT DIAMOND

Allaire – or should I say “(Nasdaq: ALLR)” – continues to rise. For those of you that don't follow the stock market, Allaire Corporation recently reported record financial results for the second quarter ending June 30, 2000. Revenue was \$33.3 million, an increase of 155% over the revenue of \$13.1 million for the same period in 1999, and an increase of 25% over the revenue of \$26.6 million for the quarter ending March 31, 2000.

The questions that probably come to mind are: What exactly does this reflect? And why should the average developer even care? For starters, you want the business that promotes your product to be in good health.

Anyway, it's an indication that a lot more people are using ColdFusion and the rest of Allaire's tools. A few of the companies that have closed deals and started using it are Barnes & Noble, Bell Atlantic, Hitachi, Johnson & Johnson, Sony, UPS, U.S. Postal Service, Virgin and Xerox. These are just a few of many, but they clearly demonstrate that Allaire's stock in the tech world isn't just rising literally, it's rising figuratively as well.

## Intel and Allaire – A Perfect Marriage?

Looking at recent industry news, Allaire is now working closely with Intel to optimize ColdFusion for the Intel Pentium Xeon processors. The joint effort of these two is designed to deliver a generation of large-volume, transaction-intensive Web applications. Using the Intel Application Solution Centers, Allaire and Intel developers were able to improve ColdFusion performance significantly on Intel processor-based Web servers. Intel has also joined the Allaire Alliance program and is now an Allaire strategic technology partner as well. The promise of this new partnership: using Intel's servers, developers will benefit through CF's increased ability to handle massive amounts of visitors with less downtime. Two very good things! Time will tell how accurate a statement that is.

## Announcing CFDJList

Now some news from closer to home. We at **ColdFusion Developer's Journal** recently announced our very first mailing list – CFDJList. Its purpose is for any and all technical discussions, CF programming- and industry-related questions, rants, raves, thoughts and ideas. It's also a great place for discussions of articles in **CFDJ**. Currently, there are no boundaries on the list, except that all discussions should be related to CF in one way or another. In its first few days (as of this writing in early August) hundreds of developers have already registered!

What's special about CFDJList, which makes it unique among other similar technical discussion lists out there, is that CFDJList is populated with our complete staff of writers and editors (and me too!), who join in the discussions, answer questions and take part in the community environment we're seeking to create. This was an often-requested online feature in feedback we've been receiving, and we're pleased to make it available at this time.

On the same note, we've launched mailing lists as well for **SYS-CON Media's** entire family of publications, so if you're interested in Java, XML, PowerBuilder or wireless development, you might find these useful to join as well. To join the CFDJList, visit [www.sys-con.com/coldfusion/list.cfm](http://www.sys-con.com/coldfusion/list.cfm).

Looking to the future: as the list grows, we anticipate breaking CFDJList into smaller, more defined lists based on interest and volume. Exactly how that's done is up to you, the reader, so drop me a line at [Robert@sys-con.com](mailto:Robert@sys-con.com) with your thoughts.

See you online!



*Robert Diamond*

ROBERT@SYS-CON.COM

ABOUT THE  
AUTHOR  
*Robert Diamond is  
editor-in-chief of  
ColdFusion Developer's  
Journal.*



# Unlocking Verity's Potential

Using Verity's searching and indexing technology on your Web site

BY DAVID C. SMITH

Since version 2.0, ColdFusion's freely bundled Verity search engine remains one of the most powerful yet seldom exploited components of the ColdFusion server. The Verity Search'97 indexing technology incorporated into ColdFusion Server provides a means for creating collections of indexed data optimized for fast retrieval, adding enormous value to any Web site big or small.

This article demonstrates the basics of setting up a Verity search collection and how to encompass all your data – static and dynamic – into one intelligent indexing solution. In addition, it shows how to display summaries without the need for preexisting META tags, highlight keywords (users love this) and build searches within searches, and the advantages of using Verity over CFQUERY.

## The Basics

The Verity engine performs searches against collections. A collection is a special database created by Verity that contains pointers to the indexed data that you specify for that collection. ColdFusion's Verity implementation supports collections of three basic data types:

1. Text files such as HTML pages and CFML templates
2. Binary document types such as PDF and DOC (see Figure 1 for a list of all supported file types)
3. Result sets returned from CFQUERY, CFLDAP and CFPOP queries

To use Verity searching and indexing technology:

- Create a Verity collection using the ColdFusion Administrator Verity page or the CFCOLLECTION tag at runtime (see Figure 2).

You must name the collection now regardless of what you're indexing.

- Populate a collection with data using options on the ColdFusion Administrator Verity page to index specific directories (usually for static or binary pages) or the CFINDEX tag at runtime (usually for dynamic data, but can also be used for static pages or building custom Verity admin templates).
- Build search forms and indexing capability into your applications using the CFINDEX and CFSEARCH tags.

Use the guidelines in Table 1 to determine which indexing method is best for you.

## Populating a Collection Using CFINDEX

Using the Administrator to create a collection of static documents is fairly straightforward. Simply specify a directory path, whether to index subdirectories, what file extensions to index (good for filtering), foreign language (if any) and, optionally, a return URL to prepend to all indexed files.

However, since most of you will likely need to index database content, follow these steps:

1. Create the collection name on the ColdFusion Administrator Verity page; at this point it's an empty container standing by for you to input data.
2. Create a CFM template that executes any query.
3. Populate the collection with data from that query using the CFINDEX tag.
4. (Optional) Schedule a task in the Administrator that runs your

indexing template nightly to keep your collections up to date.

The code below is all you need to populate a collection you named MsgIndex (following Step 1 above) from a database of threaded discussion messages:

```
<!-- Select the entire table -->
<CFQUERY NAME="Messages"
DATASOURCE="Threads">
    SELECT * FROM Messages
</CFQUERY>

<!-- Index the results -->
<CFINDEX COLLECTION="MsgIndex"
    ACTION="UPDATE"
    TYPE="CUSTOM"
    BODY="MessageText"
    KEY="Message_ID"
    TITLE="Subject"
    QUERY="Messages">
```

The table column(s) specified in the BODY attribute are what Verity actually compares search criteria against. It may contain multiple columns separated by commas, like this:

```
BODY="MessageText,Title,Company"
```

The ACTION="UPDATE" attribute appends data to your collection if the KEY doesn't already exist. The collection's KEY is similar to the primary key in a database. Using ACTION="REFRESH" would purge, then overwrite, all data in your collection. REFRESH takes more time, but it's necessary if your rows were being updated as well as added to (e.g., if users were able to edit their messages).

### Advantages of Indexing a Data Source

The main advantage of performing searches against a Verity collection instead of using CFQUERY alone is that results are ordered by relevance; the database is indexed in a form that provides faster access; and Verity offers more intelligent search capabilities. For example, Verity can find common words, both plural and singular (this is called *stemming*). Verity also allows users to apply Boolean logic (AND/OR/NEAR-type operators), which is impossible with CFQUERY. As a general rule use Verity instead of CFQUERY when you want:

- Results returned by order of relevance (Verity offers scoring variables)
- To index textual data; Verity collections containing textual data can be searched more efficiently with CFINDEX than a database can with CFQUERY
- To give users access to data without interacting directly with the data source itself

- To enable users to search more intelligently by applying Boolean logic, proximity searches and/or stemming

### Indexing Static and Dynamic Content Together

Using CFINDEX (or via the Administrator), you may populate a collection with static pages by specifying a directory tree. Then, using CFINDEX, update the collection with query data as in the example above. You may continue to update the collection with new queries or static data at any time. In theory, a single collection could contain as much of your static and dynamic data together as you like. However, you may not process multiple queries on a single collection at the same time.

A tricky situation develops, however, when you try to output combined data from more than one table since the collection's KEY value will (usually) contain numeric IDs and not know which of your tables the ID belongs to. For example, if you index two tables – Messages and Users – in the same collection and use the primary ID as the key, then ID=50 could reference either one. Therefore, when adding data from multiple database tables to the same collection, use the CUSTOM1 and/or CUSTOM2 variable of CFINDEX to hold a description that you create. Then write conditional code so when the custom attribute is recalled, the code points the ID to the correct table variable (see Listing 1; listings for this article are on page 16).

### Displaying Search Results

Once a collection has been populated via the Administrator or via CFINDEX, create a form that passes a query parameter into the CFSEARCH tag. CFSEARCH is similar to CFQUERY in that it returns records or rows of data from a collection just as CFQUERY returns rows from a database (see Listing 1).

You can pass criteria simultaneously to multiple collections by specifying a comma-delimited list of collections. Relevance is applied to the group as a whole:

```
<!-- Passing criteria -->
<CFSEARCH NAME="search"
COLLECTION="a,b,c,d"
TYPE="simple"
CRITERIA="#Keyword#">
```

Or consider grouping your output by individual collection.

In the CFSEARCH CRITERIA attribute, if you pass a mixed-case entry (mixed upper- and lowercase), case sensitivity is applied to the search. If you pass all upper- or all lowercase, case insensitivity

is assumed.

Every search conducted with the CFSEARCH tag returns, as part of the record set, a number of result attribute variables you can reference in your CFOUTPUT:

- **URL:** Returns the value of the URL-PATH attribute defined in the CFINDEX tag that's used to populate the collection. This value is always empty when you populate the collection with CFINDEX when TYPE="Custom".
- **KEY:** Returns the value of the KEY attribute defined in the CFINDEX tag that's used to populate the collection. It can be any value you choose, usually ID when indexing a database.
- **TITLE:** Returns whatever was placed in the TITLE attribute in the CFINDEX operation used to populate the collection, including the titles of PDF and Office documents. If a title wasn't provided in the TITLE attribute, CFSEARCH returns CF\_TITLE.
- **SCORE:** Returns the relevancy score of the document based on the search criteria.
- **CUSTOM1 and CUSTOM2:** Returns whatever was placed in the custom fields in the CFINDEX operation used to populate the collection (crucial when indexing multiple databases or a database with the fields you wish to display).
- **SUMMARY:** Returns the contents of the automatic summary generated by CFINDEX. The default summarization selects the three best matching sentences, up to a maximum of 500 characters.

Supported Document Types	
Documents	Versions
<b>Text files</b>	
HTML, CFML, DBM, SOML, XML	N/A
ANSI, ASCII, Plain Text	N/A
<b>Word processors</b>	
Adobe Acrobat (PDF)	all
Adobe FrameMaker (MF)	all
Apple Words	4.2
Corel WordPerfect for Windows	5.x, 6, 7, 8
Corel WordPerfect for Macintosh	2, 3
Lotus Ami Pro	2, 3
Lotus Ami Pro-Write Plus	all
Lotus Word Pro	95, 97
Microsoft Office	95, 97
MS Rich Text Format (RTF)	1.x, 2.0
MS Word for Windows	2, 6, 95, 97
MS Word for DOS	4, 5, 6
MS Word for Macintosh	4.0, 5.0, 6.0
MS Notepad, WordPad	all
MS Write, MS Works	all
KYWrite	4.12
<b>Spreadsheets</b>	
Corel QuattroPro	7, 8
Lotus 1-2-3 for DOS/Windows	2.0, 3.0, 4.0, 5.0, 95, 97
Lotus 1-2-3 for OS/2	2
MS Excel	3, 4, 5, 95, 97
MS Works	all
<b>Presentation</b>	
Corel Presentations	7.0, 8.0
Lotus Freelance	95, 97
MS PowerPoint	4.0, 95, 97

FIGURE 1: Supported file types in Verity



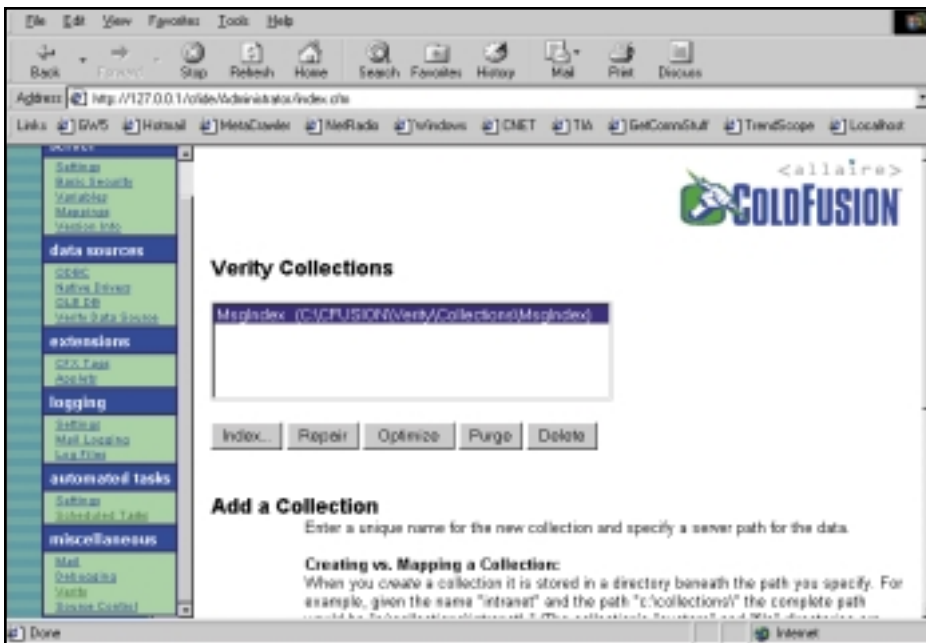


FIGURE 24 Verity collections in the CF Administrator

Using the CF Administrator or CFINDEX	
Use the Administrator if	Use the CFINDEX tag if
You want to index document files.	You want to index ColdFusion query results.
The collection won't be updated very frequently.	You need to dynamically populate or update a collection from a ColdFusion application page.
You want to generate the collection without writing any CFML code.	Your collection needs to be updated frequently.
You want to generate a one-time collection.	Your collection needs to be updated by other people.

TABLE 1 Selecting an index method

- **RECORDCOUNT:** Returns the number of records returned in the record set.
- **CURRENTROW:** Returns the current row being processed by CFOUTPUT.
- **COLUMNLIST:** Returns the list of the column names within the record set.
- **RECORDSSEARCHED:** Returns the number of records searched.

Use these attribute variables in standard CFML expressions by preceding the variable with the name of the query:

```
#search.URL#
#search.TITLE#
#search.SUMMARY#
#search.SCORE#
etc...
```

The SUMMARY attribute is probably one of the most powerful and useful attributes of Verity. This solution is perfect if you're wondering how to display useful summaries from static or dynamic pages without META tags or other meaningful abstracts built into your database content. You can always trim the summary to fewer than 500 characters by using the

MID function. For instance, if you wanted to display only 100 characters, use:

```
Mid(#search.summary#, 1, 100)...
```

In case you're wondering, there's a file under every collection called style.prm located under the Cfusion\Verity\Collections\whatever\custom\style folder. It can be opened with any ASCII editor and contains collection schema parameters. This file is used to enable/disable index schema features through macro definitions similar to those allowed by the C preprocessor. Different levels of document summarization can be uncommented in the style.prm file:

- (Default) stores the three best sentences of the document, but not more than 500 bytes
- Stores the first four sentences of the document, but not more than 500 bytes
- Stores the first 150 bytes of the document with white space compressed

### Combining Verity and CFQUERY

A powerful way to use Verity is to take search results (from a query-populated

collection) and recycle them back into a CFQUERY statement. You may want to do this to output the other fields of your table that Verity didn't index. When you populate query-driven Verity collections, specify a KEY attribute. Most of the time the KEY is the primary ID of the table. Therefore, the #search.KEY# results can be cycled into CFQUERY like this:

```
<!-- Passing criteria -->
<CFSEARCH NAME="search"
COLLECTION="MsgIndex"
TYPE="simple"
CRITERIA="#Keyword#">

<!-- query from search -->
<CFOUTPUT QUERY="search">
<CFQUERY NAME="query1"
  DATASOURCE="threads">
  SELECT * FROM Messages
  WHERE id = #search.key#
</CFQUERY>

#var1# #var2# #var3# ...
</CFOUTPUT>
```

Instead of being passed one at a time, the KEYs can also be passed in a ValueList like this:

```
WHERE id IN (#ValueList(search.key)#)
```

which would then allow you to GROUP and ORDER BY the results. *Note:* If you start grouping and ordering output from the same collection, you're logically removing the relevancy – one of the primary reasons for using Verity.

### Filtering Data

Unfortunately, ColdFusion's Verity administrator doesn't make it easy to filter out directories and files you don't want indexed. For a legacy Web site this can be a major challenge as the site's developer(s) may have kept public and private files (admin, stats, CF docs, etc.) under the same root.

One solution is to move directories and files off the root and into virtual directories. Just make sure you have any redirects set up if necessary, which can be a pain.

The other option is to delete collection records after you've indexed everything. This issue is addressed in the sidebar that contains excerpts from the Allaire Knowledge Base article #1080.

### Searches Within Searches

If a search result set returns enough records – let's say over 50 – users will usually appreciate a way to search them. Listing 2 demonstrates a simple way of doing this by populating a new Verity collection on the fly. Basically, you would:



# **Interland**

**[www.interland.com](http://www.interland.com)**

## Excerpts from Allaire Knowledge Base Article #1080

The DELETE action needs to be performed upon the collection as files are removed from the server or as records are deleted from the database so that the collection reflects the indexed entity. Below are examples of deleting records:

### Deleting a single key for a file-based collection

```
<CFINDEX ACTION="DELETE"
COLLECTION="training"
KEY="c:\training\verity\templates\germany.htm">
```

### Deleting a single key for a query-based collection

Assuming that you created a query-based collection and used EmployeeID as the KEY and one of the values of those keys is 132, your delete in this situation might look like this:

```
<CFINDEX
ACTION="DELETE"
COLLECTION="training"
KEY="132">
```

### Deleting multiple keys from a collection

This is just one example of deleting multiple keys from a collection. There may be a situation where you want to delete all keys from the collection that meet certain criteria. You

may have deleted all records from the database that have the word "mainframe" in the indexed column. If this is the case, you'd also want to remove these records from the collection. You'd first run a search. The search would return all records that meet the criteria. You would then do a delete using the word "KEY" in the KEY attribute rather than a specific key. The subsequent CFINDEX merely loops over the "query" called DeleteCertainKeys (in this example) and resolves the value of KEY (which is returned by CFSEARCH) for each record.

```
<CFSEARCH
COLLECTION="training"
NAME="DeleteCertainKeys"
CRITERIA="mainframe">
```

```
<CFINDEX
QUERY="DeleteCertainKeys"
ACTION="DELETE"
COLLECTION="training"
KEY="KEY">
```

### Deleting all keys from a collection

...Use the PURGE action [from the Administrator] or...

```
<CFINDEX
COLLECTION="training"
ACTION="PURGE">
```

1. Create a new collection in the Administrator for holding temporary data called tempCollection.
2. Output the KEY in a hidden form field after each primary result returned.
3. When the user hits a button to perform a secondary search, pass those hidden fields to another query that passes the query results into a CFINDEX tag with the ACTION="Refresh" and COLLECTION="tempCollection". This will now populate the collection in Step 1 on the fly.
4. Output the secondary search keyword into a CFSEARCH tag that's connected to the freshly populated collection from Step 3.

Understandably, this process can be somewhat system intensive if repeated over and over and large result sets are being passed. Therefore, it's a good idea to specify a maximum record set that, when reached, asks the user to perform another primary search. This on-the-fly secondary search process can be repeated down to the third level, fourth level, and so on. You can keep recycling the IDs in hidden fields.

## Highlighting Keywords

This works particularly well on the SUMMARY output for either static or

dynamic records. Basically, you build a regular expression that replaces any instance of the query keyword with a new highlighted instance. In the example below:

**Step 1:** Establish a stylesheet in the document HEAD to display the font with a yellow background:

```
<STYLE TYPE="text/css">
font.hl {background-color: yellow}
</STYLE>
```

**Step 2:** Set your new output field to newSummary.

**Step 3:** Use Replace to replace the keyword in the current summary field with your highlighted version:

```
<CFSET newSummary =
#Replace(#search.summary#, "#keyword#",
"<font class=hl><b>#keyword#</b></font>",
"All")#>
```

To take this one step further, pass the keyword variable into the URL so when users click through, the following page will also have its query text highlighted. For query-driven data it's a matter of replacing the text for your output field. For static pages you may need to read in the page via CFHTTP so you can manipu-

late the text as you read it back via #CFHTTP.FileContent#.

## Performance Considerations

Always optimize your collections, either via the Administrator or, preferably, immediately after using CFINDEX:


```
<!-- Index the collection -->
<CFINDEX COLLECTION="MsgIndex"
ACTION="UPDATE"
TYPE="CUSTOM"
BODY="MessageText"
KEY="Message_ID"
TITLE="Subject"
QUERY="Messages">

<!-- Then optimize -->
<CFCOLLECTION COLLECTION="MsgIndex"
action="OPTIMIZE">
```

Optimizing collections will significantly increase the performance of keyword searches on your site. On larger collections (e.g., 3,000-plus records) the difference can be up to seconds. Check by turning on debugging for your site to show processing time before and after optimizing.

RAM use is another consideration. From Allaire Knowledge Base Article #3690, Verity support states that "the memory requirement for a small installation using IIS (small being about 20 queries per minute and fetching HTML documents) is 64 Megs." If you plan on running lots of Verity-driven searches, plan on the extra RAM consumption.

## Conclusion

While other search engine technology exists, such as Infoseek's Ultraseek Server (expensive), Netscape Server's built-in engine or freeware Perl scripts, ColdFusion's freely bundled Verity search technology will be as easy to understand and seamless to implement as the rest of your ColdFusion applications. Remember to optimize your collections and be aware of RAM. There's good system documentation but not a lot of support in the forums. Like most things CF, you can be up and running with Verity in seconds. It's powerful and flexible, and, when properly implemented, your Web site users will praise you for making their life easier. 

**Source code is continued on pg 16.**

About the Author

David C. Smith is the Webmaster and manager of Internet development for the Telecommunications Industry Association (TIA) in Arlington, Virginia, and the lead developer behind TIA's new B2B portal, GetCommStuff.com.

||| dcsmith2@hotmail.com |||

# **Corda Technologies**

**[www.popchart.com](http://www.popchart.com)**

# **macromedia**

**[www.macromedia.com](http://www.macromedia.com)**



# **macromedia**

**[www.macromedia.com](http://www.macromedia.com)**

# If You Meet the Buddha on the Road, Kill Him

## How to become a better ColdFusion developer

BY  
HAL  
HELMS



The title of this article refers to an ancient Zen Buddhist saying, meant – like other non-sense proverbs in this tradition – to startle the student into a new way of looking at things.

“Slaying the Buddha” means to become independent, to take responsibility for what you are and learn and become. Thus, if you find the Buddha on the road – not inside you – you know this Buddha is an imposter, an illusion, the Wizard of Oz in eastern dress.

These are important ideas for developers to consider. ColdFusion’s success has been due in great measure to Allaire’s placing the power to participate in the Internet age in the hands of nonprofessionals. In that sense ColdFusion’s success has been the result of empowering people. However, I find myself noticing that as the ColdFusion community matures to the point where we have experts and gurus, some people are giving up their power, looking to the more experienced developers as judges rather than teachers, as arbiters instead of aides.

I receive a good deal of e-mail from people looking for help on their path to becoming developers. Some are looking to me to provide the “right” way to do things. The most palpable examples are those that provide some code, then ask, “What will happen if I run this?”

It’s a humorous example, I know. “Just run it!” I want to shout, but it shows how developers’ inventions and imaginations can become trapped as they look for some authority to point the direction. I understand why this happens. Development is difficult. It requires logical rigor and careful attention to detail – and even then, risk abounds. This leads people to crave the security of an expert to assure them, “This is the right way.”

Developers can recover their

own power and use the experts among us as colleagues who may be able to provide assistance, but never dictate. You can, in essence, become your own guru. Here are some ideas about how to do this.

### 1. *Read – A Lot*

This magazine is a good example. It’s amazing how many people who say they want to learn don’t avail themselves of the available resources. I recognize I’m preaching to the choir here as you’re already reading this magazine, but many of our fellow developers aren’t. Why? “Too expensive,” many of them tell me. I tell them: “You’re in a high-paying profession. If you get only one idea for the entire year that helps you become a better developer, how many thousands of dollars is that worth?”

There are other excellent resources, beginning with the documentation Allaire provides. When I began learning ColdFusion, I pored over the language references. Then, when I needed to do something, I often had a vague memory that there was some tag or function that could help with this. There are some excellent sites on the Web with ColdFusion tips and tutorials. Why not frequent these to see how other developers approach problems.

In addition to Ben Forta’s excellent books on ColdFusion, others are available to help you. Not all books need to be on the subject of ColdFusion proper to help you become a better programmer. There are many excellent books on programming that cover such non-ColdFusion topics as XML, debugging, SQL, database design and object-oriented

programming. Again, I’m looking for one good idea that I can use per book. Anything more than that is an unexpected bonus.

I read books on a wide variety of topics and continually find that disciplines seemingly unrelated to programming often offer insights that help me address some aspect of development, or simply expose me to new ways of thinking about things – one of the most essential and elusive of skills.

### 2. *Try Things Out*

One thing I like about a newsletter is that it makes me take a subject and explore it in detail. You can do the same thing. Take a topic – arrays, structures, WDDX, whatever – get out your ColdFusion Studio and try things out. Play with it.

If you watch young children play video games, you’ll see the amazing power of play. I had this experience with my own son. When he was 5 I bought him Super Nintendo and some games, with some trepidation. I thought we’d be able to play together. A nice thought, but within two weeks he was so far ahead of me that “playing together” was really an exercise of charity on his part as he watched me fall farther and farther behind. I experienced the ultimate humiliation when he tried to cheer me up: “Hey, you did really well...for an adult.”

I have a directory on my computer called *Test* that I use to try things out. Do you need to pass an array from one page to another? Thinking about using WDDX? Then try it out! See how long it takes to translate it to an XML string. How easy is it to store? Confused by

# DigitalNation

**[www.dedicatedserver.com](http://www.dedicatedserver.com)**

These forums at [forums.allaire.com](http://forums.allaire.com) can be a tremendous source of help with thousands of ColdFusion developers available to lend a hand. However, a few words are in order when using the forums. Just as we've all seen how the medium of e-mail can transform courteous, civilized people into ravenous flame-beasts, the forums are best used if we can avoid sloppy posts and adhere to an etiquette that provides the best environment for people to

- **Cross-posting** You've seen those posts that appear in every possible topic. This is really a violation of good manners. Consider how

I hope you'll find some of these ideas to be helpful along your path to becoming a better, more skilled ColdFusion developer. 🍀



```

Listing 1
<!-- Passing criteria --->
<CFSEARCH NAME="search"
COLLECTION="MsgIndex"
TYPE="simple"
CRITERIA="#Keyword#">

<!-- Output conditional results --->
<html>
<body>
<CFOUTPUT QUERY="search">
<CFIF #search.custom1# is "forum">
<a href="threads.cfm?ID=#search.key#">#search.title#</a><br>
#search.summary#<br>
<CFELSEIF #search.custom1# is "user">
<a href="userDetail.cfm?ID=#search.key#">#search.title#</a><br>
#search.summary#<br>
<CFELSE>
<a href="#search.url#">#search.title#</a><br>
#search.summary#<br>
</CFOUTPUT>
</body>
</html>

```

---

```

Listing 2
<!-- evaluate whether the user has passed a secondary search
parameter --->

<CFIF IsDefined("form.keyword2")>

<!-- query from hidden field IDs --->

```

The code listing for  
this article can also be located at  
[www.ColdFusionJournal.com](http://www.ColdFusionJournal.com)



# **CFX HOSTING**

**[www.cfxhosting.com](http://www.cfxhosting.com)**

# Just Your Type



BY  
BEN  
FORTA

ColdFusion features support for several different types of variables you can use in your applications...some types are simpler than others to use.

When determining the type to use, simplicity and ease of use shouldn't be the only deciding factors. Use the wrong type...and performance can suffer.

## Simple Variables

Let's start with the basics. More often than not you'll find yourself using simple variables. There are essentially three ways to create such variables:

1. The `<CFSET>` tag is used to create or set (overwrite) a variable.
2. The `<CFPARAM>` tag is used to create a variable (with a default value) only if that variable doesn't already exist.
3. Assignment within a `<CFSCRIPT>` block is used in much the same way as the `<CFSET>` tag.

Simple variables can exist within many different scopes, but regardless of the scope used, simple variables are just that – variables that contain simple single values like strings, numbers and dates. The following code snippet creates a simple variable:

```
<CFSET yesterday=DateAdd("d", -1,
Now())>
```

To display (or access) simple variables, you just need to refer to it, as follows:

```
<CFOUTPUT>#DateFormat(yesterday)#
</CFOUTPUT>
```

## Lists

Lists are strings that contain one or more elements separated by a specified delimiter. A comma-delimited list of values is obviously a list. A sentence is a list of words delimited by spaces (or other punctuation

characters). Even phone numbers can be thought of as lists delimited by hyphens or parentheses.

Lists are actually not a variable type; they're really just simple variables. In truth, any variable can be treated as a list. There are no special functions for creating lists. You'd create one just as you would any simple variable. The following code snippet creates a list of four product IDs:

```
<CFSET prods="123A,76GH,901HY,91AA">
```

The following code snippet also creates a list, a space-delimited list of words:

```
<CFSET title="ColdFusion Developer
Journal">
```

Accessing specific list elements requires the use of special func-

tions, all of which begin with the word *List*. `ListFirst()` returns the first element in a list, `ListLast()` returns the last and `ListGetAt()` can be used to obtain any list element by specifying its position. You can even sort, search and edit list elements, all using `List` functions. But you can't access specific list elements directly – for that you must use special functions. To display the third element in the above list you could use the following code:

```
<CFOUTPUT>#ListGetAt(prods,
3)#</CFOUTPUT>
```

The default list delimiter is a comma, but all `List` functions take an optional attribute that allows you to specify an alternate delimiter. For example, to count the number of words in a string, you could use the following code:

```
<CFSET num_words=ListLen(sentence, "
")>
```

Lists are easy to use, and they're especially convenient when working with form field values and SQL statements (both of which also use comma-delimited values). But list access comes with a significant performance overhead. Because lists are actually strings, requesting the third element as we did above requires ColdFusion to parse the string to find that element. ColdFusion can't access that element directly. It must first go through that parse step each time that element is accessed.

## Arrays

Arrays are special variable types, usually used to contain related data in a structured and sequential man-

“”

Lists are  
actually not a  
variable type;  
they're really  
just simple  
variables

# **Virtualscape**

**[www.virtualscape.com](http://www.virtualscape.com)**

ner. An array is a set of placeholders, each containing a value. And arrays can be multidimensional. ColdFusion supports up to three dimensions, though you'll rarely have use for three-dimensional arrays. A list of values could be stored in a one-dimensional array, while a shopping cart (which needs to contain multiple items, each with multiple properties) could be stored in a two-dimensional array.

Arrays are manipulated using special functions that all begin with the word *Array*. They're created using the `ArrayNew()` function, which takes a parameter that specifies the number of dimensions needed. To create a SESSION variable shopping cart as a two-dimensional array, you could do the following:

```
<CFSET SESSION.cart=ArrayNew(2)>
```

Array elements can be accessed directly by specifying the appropriate index (position). `Array[1]` refers to the first element in a one-dimensional array, `Array[2][3]` refers to the third column in the second row in a two-dimensional array (of course, you'd need to use the actual array name, not the word *Array* as I did). So to assign a value to our shopping cart, you could do the following:

```
<CFSET SESSION.cart[2][3]=100>
```

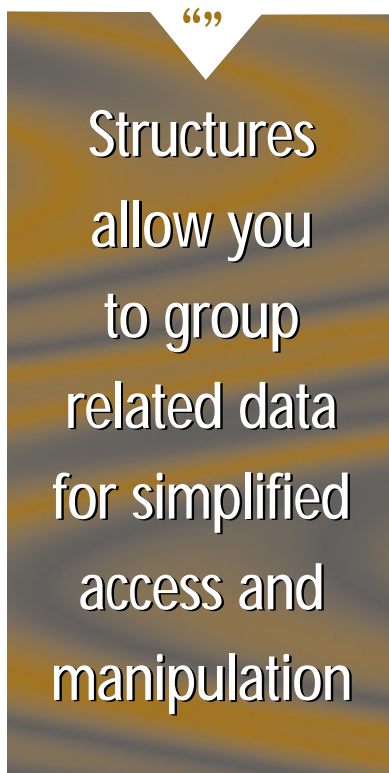
To display an array element, you simply refer to it, like this:

```
<CFOUTPUT>#SESSION.cart[2][3]#
</CFOUTPUT>
```

Arrays are extremely useful when working with sets of related data, so ColdFusion includes an extensive set of array manipulation functions that do things like sorting, finding high or low values, and much more. And array access is fast. ColdFusion treats arrays like sets of values (unlike lists), and as such is able to access specific array elements without having to perform unnecessary processing or intensive parsing.

It's worthwhile to note that two-dimensional arrays are usually thought of as grids, although this isn't technically accurate in ColdFusion. Unlike most other languages,

ColdFusion's multidimensional arrays are actually arrays of arrays. The difference is subtle. In traditional arrays, if the first row has six columns, then all rows will have six columns (I'm using the terms *rows* and *columns* for simplicity's sake only). In ColdFusion arrays grow dynamically and as needed, as do arrays within arrays. So the fact that one row has a certain number of columns doesn't mean that other rows will have the same number. As such, you must always make sure



that specific index elements exist before you attempt to use them.

### Structures

Structures are one of the most powerful and flexible variable types in ColdFusion. They can be thought of as variables that contain other variables. They have two primary uses:

- Related data can be grouped together for simpler processing.
- Array-style data can be manipulated using element names (called *keys*) instead of numeric indexes.

There's so much to say about structures that the subject would probably best be served by a column unto itself, but here's the basic

concept. Suppose you were working with customer information. You'd need a variable for customer ID, a variable for first name, a variable for last name, and so on. All those variables together make up a set of data, but as separate variables there's no programmatic relationship between them. So you couldn't loop through all the variables without hard-coding their names, nor could you pass the entire set to a custom tag without listing each individually. Structures allow you to group related data for simplified access and manipulation. So instead of variables named "id" and "first\_name", you'd have variables named "customer.id" and "customer.first\_name".

As you'd expect, structures are manipulated using CFML functions that all begin with the word *Struct*. To create a new structure, you'd do the following:

```
<CFSET customer=StructNew(>
```

To set structure elements, you could do the following:

```
<CFSET customer.id=100>
<CFSET customer.first_name="Ben">
```

To display a value in a structure, you simply refer to it by name, like this:

```
<CFOUTPUT>#customer.first_name#
</CFOUTPUT>
```

You can loop through structures and pass them as tag attributes, and more. In fact, as of ColdFusion 4.5, many of the internal sets of values are actually structures. For example, a structure named *FORM* contains all form fields, and a structure named *URL* contains all URL parameters. Thus, when you refer to *FORM.name*, you're actually accessing a structure member (perhaps without even knowing it).

Like arrays, ColdFusion can access structures quickly and efficiently. And like arrays, structures can contain other variable types too (like other arrays or structures). As such, structures are ideally suited for grouping related data, as well as for organizing complex or large data sets. Structures are particularly



well suited for use with SESSION and APPLICATION variables. Instead of having lots of what are essentially global variables floating around, you'll perhaps have just one structure that contains all the other data.

### Picking Your Type

Lists are easy to use, but ColdFusion processes them far more slowly than it does arrays or structures. While I haven't seen any formal numbers from Allaire on the difference in performance, my own testing (creating massive lists, arrays and structures, then extracting thousands of randomly selected elements from them) indicates that list access is usually as much as three times slower than array or structure access.

Lists are easy to use. If you need to build dynamic SQL WHERE clauses, lists are a must. If you work with form fields that have multiple values, lists (and the List functions) are invaluable. But lists are not well suited for any complex data or more complex processing. That's not

what they were designed for. The more you access list members, the more benefit there is to be gained from using arrays or structures instead.

Arrays are ideal for sets of data that can be accessed by index. Structures are ideal for grouping any related data (and structures can also be used as associative arrays – arrays accessed by key name rather than by index). Both arrays and structures can contain other arrays and structures, making them as powerful and flexible as they are efficient. But arrays and structures aren't as simple to use as lists.

Some applications are best served by a combined approach. For shopping carts I like using a structure containing an array of structures (within the SESSION scope). One application I developed has a structure called SESSION.shopping. It contains a couple of variables pertaining to the customer (things like customer ID), as well as an array called *cart* that is the cart itself. The array contains structures, one per

item in the cart. So the customer ID can be accessed as SESSION.shopping.id, the cart as SESSION.shopping.cart, the third item in the cart as SESSION.shopping.cart[3] and the quantity associated with that item as SESSION.shopping.cart[3].quantity.

### Summary

With ColdFusion there's always more than one way to do just about anything. You're the one who has to decide what solution will work best for you. Inevitably, making decisions involves trade-offs. Often the simplest option isn't the most efficient – but on the other hand, there's something to be said for simplicity.

The bottom line is this: all of ColdFusion's variable types are useful, and all have their strengths and weaknesses. Your job is to understand these strengths and weaknesses so the design decisions you make are educated and informed ones.



### ABOUT THE AUTHOR

*Ben Forta is Allaire Corporation's product evangelist for the ColdFusion product line. He is the author of the best-selling ColdFusion 4.0 Web Application Construction Kit and its sequel, Advanced ColdFusion 4.0 Development, as well as Sams Teach Yourself SQL in 10 Minutes. He recently released Allaire Spectra E-Business Construction Kit, and is now working on books on WML and JSP.*

BEN@FORTA.COM

# Intelligent Environments

[www.screensurfer.com](http://www.screensurfer.com)

# eWebEditPro 1.8.0.2 from Ektron

How can you gain time for programming?  
Have your contributors edit HTML  
content themselves

REVIEWED BY  
MICHAEL  
SMITH



eWebEditPro from Ektron is a browser-based WYSIWYG editor that lets your Web site users create HTML content even if they don't know HTML.

In this review we discuss why you'd need eWebEditPro, and point out some of its features and limitations.

## Going in HTML Edit Circles?

How many times have you brought up a newly created Web page in a browser window and spotted some content you'd like to change? If you're a developer, you can go to your HTML editor or ColdFusion Studio, make the change, save it, go back to your browser and reload. Still not quite right? Another round-trip to the editor is required. For users it's even more complex. They have to contact the person who maintains the page, communicate the change and hope they get it right. Often users will put up with "good enough" to avoid the added time and bother of another edit cycle.

To prevent your whole life from turning into an endless stream of content change requests, you write your content into a database using memo fields and display it using ColdFusion. You even write a simple admin interface with an edit window using <TEXTAREA> tags so your company's content managers can edit the HTML themselves. Problem solved? Not quite. Not only do your users still have to do the edit-save-view cycle to test their HTML, they must also know HTML and be perfect at matching opening and closing tags. Not too likely.

## <TEXTAREA> on Steroids

There's a better solution – eWebEditPro – a WYSIWYG (What You See Is What You Get) editor for the Web. With eWebEditPro your page content appears in an editor window that has controls similar to those in Microsoft's WordPad. Plus your users can perform spell checks; insert bookmarks, hyperlinks and pic-

tures; and insert and format tables (see Figure 1). This tool also allows your content contributors to cut and paste text from Microsoft Word or Excel. Now users can edit Web pages directly!

On its most basic level, eWebEditPro is a friendly replacement for the <TEXTAREA> tag. In reality, it's more than that. For example, eWebEditPro provides a familiar, Microsoft Office-like interface for editing and adding text and graphics. It'll even import and clean up Word and Excel files, leaving out those annoying extra HTML tags so common in Word's HTML export.

Although we only tested it with ColdFusion, the documentation states that eWebEditPro can be used with ASP or even some of the popular content-management packages such as Vignette. Unfortunately, the process of using the software with other platforms appears to be more difficult – although Ektron indicates it's working toward enhancing this. We found the software to be stable and the features worked as stated. We pasted in an entire Word 2000 document. Except for a few minor problems, the formatting was true to the original – even the images came in. If you don't like using the WYSIWYG interface, a simple right-click allows you to reveal and edit the actual HTML code.

The following is all you need to use the editor in your application in place of TEXTAREA fields.

```
<CF_WebEditorPro FormName="frmDoc-
Maint" ReturnField="TextHTML1"
License="www.mycompany.com?123456"
Buttonshyperlink="False"
HTML="#my_query.press_release_body#">
```

## Customization

In ColdFusion you can control such code elements as fonts, sizes

## VITALS

**eWebEditPro:** Ektron, Inc.

**Address:** 5 Northern Blvd., Suite 6  
Amherst, NH 03031

**Web:** www.ektron.com

**Phone:** 603 594-0249

**Fax:** 603 594-0258

### Test Environment:

**OS:** Win 98, Win NT

**Web server:** IIS Apache

**CF:** 4.5.1 Ent

**Browser:** IE4, NS 4.7

**Pricing:** \$299 for each copy licensed to a base URL with 10 author/editor seats. Additional 10-seat blocks available for \$299.

and colors on your Web site. It even lets you turn off features you don't want your users to have access to, such as HTML "reveal codes." You can either customize permanently via an XML document or specify options on the fly. For instance, adding the Buttonshyperlink="False" directive in the example above turns off the hyperlink button on eWebEditPro's interface. You have similar control over every button. Images can be stored in a library, and the administrator can limit users to display only pictures that are already there. Or you can allow users to upload their own images.

## Installation

Installation of version 1.8 was a snap – a big improvement over earlier versions, which had to be installed manually. ColdFusion and ASP server-side installations are available for Windows. InstallShield prompts you for the install location, which should be a path on your Web site. It'll also ask you for the location of your ColdFusion custom tag folder. Note that we tested

# **The Short List**

**[www.theshortlist.com](http://www.theshortlist.com)**

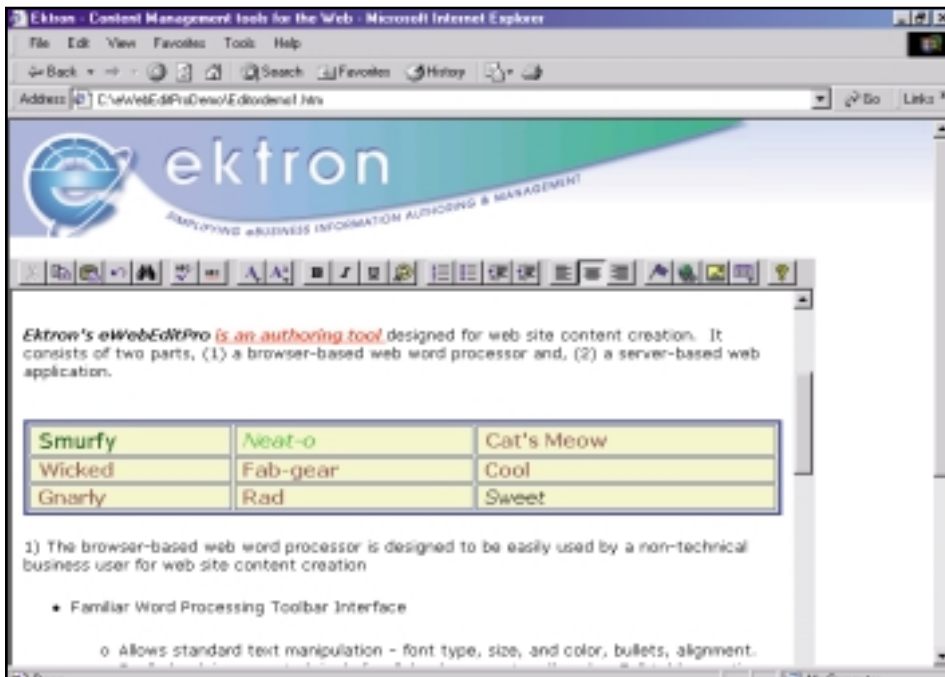


FIGURE 1: eWebEditPro in action – having fun with the Insert Table function of the Editor

eWebEditPro with both IIS and Apache, but only the former is mentioned in the documentation.

The big change for version 1.8 is that Netscape 4.7 on Windows is now supported (Internet Explorer 4.01 or later must be installed).

A client-side install is included in the download package and copied to the install location. In the Netscape version this installs the Esker ActiveX Plug-in into Netscape on the client computer. This is necessary because eWebEditPro is an ActiveX component that must be downloaded and installed on the end user's machine. It's not a problem since the download is an automatic process and the file is only 150K. On our test computer this download took less than a minute and was done only once. To use the software, the user *must* have Windows with Internet Explorer 4.0. If a user attempts to edit your site with an unsupported browser, it simply degrades itself to a regular textbox. There are no server-side requirements.

The Ektron code indicates that eWebEditPro limits text fields to 65,000 characters, though in practice browser and ODBC limits may be smaller.

## Tech Support

We contacted tech support for a problem encountered when cutting

and pasting a Word 2000 document into an empty page. It produced what appeared to be a double-spaced page. Then, when performing a cut-and-paste of a Word 2000 document into an existing Web page and choosing the Clean HTML Code option, the font size of the headings was considerably reduced and the first paragraph was missing.

The response from Ektron's support staff was exceptional. A knowledgeable support person answered the phone on the first ring. He explained that the word processing CR/LF commands are converted into <P> tags instead of <BR> tags, so some cleanup of cut-and-pasted text may be necessary. The font size changed because all font information is retained in a stylesheet, which can't be copied from Word. If no font information is available, the content defaults to the browser default font type and size. FrontPage also has this "feature."

Although no solution could be immediately given for the lost paragraph, tech support requested the Word document be forwarded to him. We obtained a response a few hours later with a request to reproduce the situation. Fortunately it couldn't be reproduced.

Tech support is available by e-mail at support@ektron.com.

## Licensing

To work on your ColdFusion Web site, you need a license key issued by Ektron. These are issued manually, but an automated process is in the works. The license key is in the form of \*.mywebsite.com?1234567890. The license parameter of the <CF\_WebEditorPro> tag must contain this value each time it's used. This key is good for any sub URL of the main URL such as www.mywebsite.com/news. If an IP address is used in place of your domain name, it won't work unless you have a license key for the IP address. A quick call to Ektron revealed that you don't have to purchase the extra license for an IP as long as that IP address is associated with the domain name. And the purchase price for the production server generously includes free licenses for development, testing and staging servers! For \$299 you get a copy of eWebEditPro for use within one domain for 10 discrete *named* users. Each additional 10 users costs another \$299. These are admin users – you can have as many viewers of your site as you like.

We highly recommend you download the free demo from their site. The demo is fully functional and is provided with a 30-day evaluation license key. That way you can try out eWebEditPro on your own site. Once you've tried it I predict you'll never go back to an old-fashioned <TEXTAREA> tag!

## Conclusion

eWebEditPro saves content as HTML that can be displayed in ColdFusion Web pages without the editor being present. The WYSIWYG Web editor is a great tool for any content management team or individual user who requires easy editing of formatted text but has little or no knowledge of HTML.

## Acknowledgment

I'd like to thank fellow MDCFUG member Eron Cohen and Roy De-Jarnette, a senior developer for TeraTech, for their help in testing eWebEditpro.



MICHAEL@TERATECH.COM

## ABOUT THE AUTHOR

Michael Smith is president of TeraTech ([www.teratech.com](http://www.teratech.com)), a consulting company based in Rockville, Maryland, that specializes in ColdFusion, database and Visual Basic development. Michael also runs the MDCFUG and CFUN2K event. Reach him at (800) 447-9120, x500.



# Shift 4

**[www.shift4.com](http://www.shift4.com)**

# Permission Framework

## Add a flexible access control scheme to your Web site



BY  
**KELLY  
BROWN**

Most commercial Web sites have secure areas that are accessible only to authorized users. Here's one schema for managing access control for a Web site.

Many times you need to limit access to particular templates on your site, either for customers who must purchase enhanced capabilities or for administrative functions available to a select few. The framework described in this article gives you a flexible access control scheme that can easily be added to your Web site.

## Database Design

Like all good dynamic systems, this one starts with the database design. It boils down to two simple concepts: users and permissions, where permissions regulate access to various features. We'll have two main tables in our schema: **USERS** and **PERMISSIONS**.

The `USERS` table contains information about the users such as their login name and password.

The PERMISSIONS table contains a list of permissions that govern which features a user may or may not access. Each user can be assigned numerous permissions so we need an additional table, USER\_PERMISSIONS, to represent this many-to-many relationship (see Figure 1).

This structure gives us a great deal of flexibility. We may only want to define permissions to distinguish between administrative and normal users, or we may want a finer-grained approach with permissions for each ColdFusion template on our site. With this database design we can define as many permissions as we need.

## Using Session Variables

We've defined our permissions, but how do we use them? This is

where our old friend the session variable comes into play. Typically, when you log in we set a flag in a session variable to track your login. We'll add an additional variable to your session that keeps a list of your permissions. Once this session variable is set, we can check it in our templates to see if you have access.

## Code Walk-Through

The two steps to implementing this permission framework in ColdFusion are:

1. Setting the permissions as part of the login process
2. Checking the permissions in our templates

Listing 1 provides a sample template to process a login. When you successfully log in with a user name

### Listing 1: Processing Login and Setting Permissions

```
<!-- Process login form -->
<cfquery name="check" datasource="security">
  SELECT user_id
  FROM users
  WHERE user_login='#FORM.login#'
    and user_password='#FORM.password#'
</cfquery>

<!-- If user found set session variable -->
<CFIF check.RecordCount IS NOT 0>
  <CFSET Session.user_id=check.user_id>
<!-- Get list of permissions -->
<CFQUERY Name="permissions" Datasource="security">
  select permission_name
  from permissions
  where permission_id in
    (select permission_id
     from user_permissions
     where user_id=#check.user_id#)
</cfquery>

<!-- Build a list of user permissions -->
<CFSET Session.permissions="">
<CFLOOP QUERY="permissions">
```

```
<CFSET Session.permissions =  
    ListAppend(Session.permissions,permission_name)>  
</cfloop>  
<!--- Finished login, send to main menu --->  
<CFLOCATION URL="menu.cfm">  
</CFIF>  
  
<html>  
<head>  
    <title>Invalid Login</title>  
</head>  
  
<body>  
    <B>Invalid Login.</b>  
    Use the back button to try again.  
</body>  
</html>
```

## CODE LISTING

DDDDDDDDDD

The code listing for  
this article can also be located at  
[www.ColdFusionJournal.com](http://www.ColdFusionJournal.com)

and password, we retrieve your permissions from the database and add them to your “permissions” session variable. You’ll notice that we’re placing the permission name in the variable, not the permission ID. We could use the ID, but it’s hard to remember what permission 14 is. It’s easier to work with permission names in your templates, such as “Create User” or “Sales Report.” Use easy-to-understand permission names as they’ll be easier to work with later on.

Checking for permissions in our template is straightforward. We examine the list of permissions in your session variable to see if it contains the one needed to access this template. If the permission isn’t found, we display a message stating this, then we stop the remainder of the template from running.

```
<CFIF ListFind(Session.permissions,
"Admin") IS 0>
    <CFINCLUDE TEMPLATE="NoAccessMes-
sage.cfm">
    <CFABORT>
</CFIF>
```



FIGURE 11 ER diagram for security database

We can also use the permissions in other ways. For instance, we may not want to display links to pages you don’t have permission to access. In the following example we’ll show the link to the administrative report only if you have the “Admin” permission.

```
<CFIF ListFind(Session.permissions,
"Admin") IS 1>
    <A HREF="AdminReport.cfm">Adminis-
trative Report</A>
</CFIF>
```

## Conclusion

This framework is straightforward and can easily be incorporated into most sites. It gives you the flexibility to have simple or complex permissions as needed, and can be used in many types of sites from an intranet with administrative privileges to an e-commerce site in which customers can purchase additional permissions. Try it out!



**ABOUT THE AUTHOR**  
*Kelly Brown is the chief technology officer of About Web (www.aboutweb.com), an Internet solutions provider in the DC area. He has a BS and MS in computer science and is a Microsoft certified system engineer.*

KBROWN@ABOUTWEB.COM

# Virtual Scape

[www.cirtualscape.com](http://www.cirtualscape.com)



BY RAYMOND H. THOMPSON

# Got a Problem with Duplicate Records?

**Maybe it's the multithreading in the CF server**

**T**he thrust of this article is the multithreading environment in ColdFusion. Multithreading is nothing more than the ColdFusion engine spawning more than one copy of a given task.

For most Web applications this is a good thing as it improves the processing speed of the Web site. But nothing is free, and there's a significant "gotcha" that will occur in ColdFusion applications. There's a way around this problem, but in my opinion it's something that needs to be changed by Allaire.

The information in this article is specific to version 4.5.1 of the Enterprise version of ColdFusion. It also deals with a remote database connection to Oracle using native and ODBC drivers via SQL\*NET. Other configurations may not suffer from the same problem.

# **Allaire Developers Conference**

**[www.allaire.com/conference](http://www.allaire.com/conference)**



## The Setup

So what is this multithreading thing and why is it good and desirable? Multithreading is really nothing more than a task, such as ColdFusion, spawning another copy of some portion of itself in separate memory space. The operating system will manage this as a separate entity. Crashes in this separate task will not affect the parent task.

This is really a good thing. Imagine that there's some significant error in the code and that this error will generate some really long script times (a loop that never ends). Having been both the initiator and the recipient of such code, I know its affects can be devastating. While this errant code is executing, no other scripts can be processed. This effectively locks the site until the script is terminated. Now imagine the ColdFusion administrator has been set so as not to time out tasks that take an excessive amount of time. The site is basically toast and the ColdFusion services must be recycled. This is generally not received well by customers or clients. It might be okay in development, if the developers are willing to suffer the wrath of other developers.

To avoid this difficulty, the ColdFusion server will start another task, or thread, that will handle the request for each request that arrives. With each request having its own separate space, catastrophic errors in one script won't cause others to become corrupted. This also allows scripts to operate simultaneously on systems that have multiple processors, which provides a real performance boost. Each site, via the ColdFusion administrator, can limit the number of simultaneous requests.

But nothing is free. Problems with this multithreading will cause difficulty in some scripts. Most affected are scripts that post items to a database. Without

some coding considerations some nasty surprises will appear.

So what is this problem and how is it dealt with? The problem is that ColdFusion cannot, or rather, does not distinguish between requests from different clients. This is worth restating. The ColdFusion server doesn't know if the request is coming from a client that already has a request running or from a different client.

Again, so what? Well, submitting a form to update a database, depending on how the database is being updated, can cause significant problems.

## The Problem

To fully illustrate the problem, a visual aid will certainly help. You'll find it in Listing 1. Let's assume that the page submitted is updating a database. The application coded by "Joe Smart" coder is written in such a way as not to allow duplicate records in a database. Joe did this by using code that would check the database for the existence of the record to determine whether it should be updated or inserted. The code is fairly straightforward. It checks for the existence of a record based on some key. If the record count is zero, the record is inserted. Otherwise it's simply updated. This code would seem to be quite solid and will work all the time.

Note that NO <CFTRANSACTION> tag is used. This was done deliberately as the <CFTRANSACTION> has no bearing on this difficulty.

But what happens if the user clicks twice on a Submit button? Probably not much in this case as the code will execute quickly. What happens when the query takes some time, either because of other database activity or updating of many records? Since the application is running on the Web, what if the Web is slow and the first message gets hung up somewhere?

If the user clicks on the submit button twice, problems will probably appear. What happens is that the ColdFusion server executes the first request when the next request arrives. ColdFusion spawns another thread that executes the same code. Now two copies of the code are running at the same time, both updating the database.

Timing now becomes critical. If the first request doesn't find a record in the database, and starts inserting the data into the database about the time the second request starts executing, the second request won't find a record. The first request finishes inserting the record into the database; the second request then finishes and inserts a record in the database. Now there are two records in the data-

base. Unless some constraints are placed on the database, or triggers set – and this isn't always possible – there will now be two records in the database. Duplicated data.

The user eventually notices the two records. The developer is notified, and some heads are scratched as the code is examined. No way could this code cause duplicated records. This is true. But the ColdFusion server can, and did. The time chart in Table 1 indicates how.

Time Point	Request 1	Request 2
1	Checks DB	
2		Checks DB
3	Inserts Record	
4		Inserts Record
5	Request Done	
6		Request Done

TABLE 1: Time Chart

In this scenario neither request will find a record to update, so the script will insert the record.

This is one of those obscure problems that cause developers no end of problems. Chasing these types of problems is generally futile as the problem can't be reproduced easily. Code can be examined until the developers are blue in the face and the problem won't show itself.

Granted, the problem is reproducible only when the queries involved are long running and the user gets impatient. But I have this philosophy that if something in code can happen, it will. Just ask Murphy.

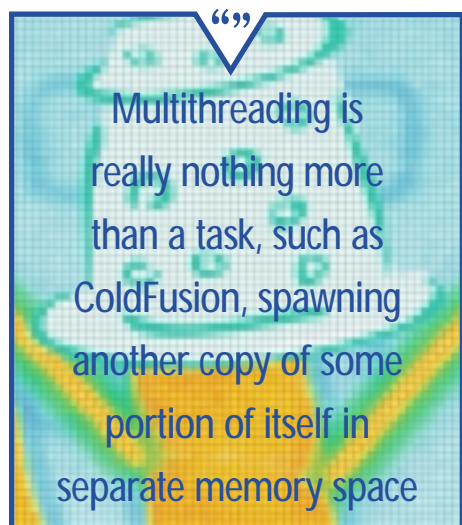
## What to Do

A couple of things can be accomplished. One is by Allaire; the other is by developers. The solutions are presented in what I feel is the order in which they need to be addressed.

Allaire needs to fix the ColdFusion server. Simple solution, and it solves a lot of application problems in one sweep. Allaire needs to modify the ColdFusion server to specifically queue requests with the same CFID and CFTOKEN. This will flatly stop the duplicate update problem. There's no intelligent reason for requests with the same CFID and CFTOKEN to be processed simultaneously. It just doesn't make sense.

This type of queuing will also improve the performance of the Web site slightly by allowing other users to get their requests processed while the impatient user simply waits. But that's only a minor advantage.

By queuing the request, the second





# XML DevCon FALL 2000

Register by  
Friday, October 6  
and  
**SAVE \$200**

**Conference:**  
November 12-15, 2000

**Exhibition:** November 13-14, 2000

DoubleTree San Jose Hotel  
San Jose, California

## Plan to Attend the Largest XML Event Coming to Silicon Valley

**...Featuring an All-Star Lineup of Instructors  
and over 100 Participating Companies...**

If there's one XML Conference you attend this fall,  
make sure it's XML DevCon -

*the only XML event to combine the most XML vendors with the most technically advanced, comprehensive technical program. Join 5,000 XML enthusiasts - the industry's most respected technical experts, sought-after gurus and advanced users. Come immerse yourself in four days of the hottest XML techniques and master new skills taught by those who are defining XML's future.*

### Event Highlights

- Back by overwhelming demand, this will be the largest XML event to come to Silicon Valley in 2000.
- Expanded program - 6 tracks, over 100+ sessions to select from.
- Special Preconference Tutorial on Sunday - choose from 11 in-depth sessions.
- Night School Sessions - select from 18 in-depth topics on Sunday, Monday and Tuesday nights.  
Extend your day program or take independently
- Learn from masters. Just take a look at the Faculty List.

### Distinguished Faculty Members Include...

Charles Goldfarb, father of SGML and author of *The XML Handbook*, *XML Web Kit*

Jim Gray, coauthor of *Transaction Processing: Concepts and Techniques* and *The Benchmark Handbook: For Database and Transaction Processing Systems*

Don Chamberlin, codesigned *SQL* and *Quilt* and authored *A Complete Guide to DB2 Universal Database*

Joe Celko, author of *SQL for Smarties*, *Instant SQL*, *Data and Databases*, and *Joe Celko's SQL Puzzles & Answers*

Kurt Cagle, author of *XML Developer's Handbook*

Michael Floyd, author of *Building Web Sites with XML*

Elliott Rusty Harold, author of the *XML Bible*, *XML: Extensible Markup Language*

Ken Holman, author of *Practical Transformation Using XSLT and XPath*

Bruce Peat, coauthor of *Professional XML*

Simon St. Laurent, author of *XML: A Primer*, *XML Elements of Style*, *Building XML Applications*, *Cookies*, and *Sharing Bandwidth*

Tommie Usdin, editor of *Markup Languages: Theory & Practice*

Paul Brown, coauthor of *Object-Relational Databases: The Next Great Wave*

### Explore Today's Most Relevant Topics Including...

- \* Bleeding Edge of XML
- \* Building Web Sites with XML
- \* XML for Programmers
- \* Introduction to XML
- \* Practical Transformations Using XSLT and XPath
- \* Integrating XML, Java and CORBA as a Distribution Mechanism
- \* XML vs EDI
- \* Enterprise Data Access and Content Syndication
- \* Using XML to Facilitate Data Flow
- \* BizTalk Development
- \* Tree-Based XML Processing with Perl
- \* Generating XML from Form Data
- \* Application Integration Using XML
- \* XML and JavaServer Pages
- \* XML Meets Middleware
- \* XOC and B2B Collaboration
- \* DOM Level 2
- \* Where's the Data? Trading Between Relations and XML
- \* OASIS Standards Activities
- \* XML APIs in the Java Software Platform
- \* Financial Industry B2B Messaging Standards
- \* Lycos Multitier XML Architecture
- \* SMIL - You are Seeing Candid XML
- \* Understanding Quilt
- \* EAI Using XML and XSL
- \* Content Generation and Distribution Using XML and Jini
- \* Integration Services, Wireless and Databases
- \* Using WebDAV
- \* Programming for Browser Interoperability
- \* XML and Open Source Medical Records
- \* Automatically Creating Servlets from XML
- \* Java, B2B, Wireless Drive eBiz
- \* Understanding SOAP
- \* Querying XML Documents
- \* Standard XML Templates for Healthcare
- \* XML and Scripting Languages
- \* VoiceXML
- \* XML and Stylesheets
- \* Structuring Documents for Databases
- \* Topic Maps, Portals and Info Management
- \* Using SAX and Java to Explore XML
- \* Building Wireless Applications Using JSP
- \* Advanced SOAP Programming
- \* Interface Development with XUL
- \* Mapping DTDs to Databases
- \* Networked Marketplaces
- \* Java-Based Xbeans
- \* 21st Century XML Servers
- \* XML-Bots
- \* XHTML: XML for Client-Side Authors
- \* Serializing Query Results in XML
- \* B2B eXchanges
- \* ebXML & JAXM-eCommerce and Java
- \* XML Support in TSpaces
- \* XML Views in SQL Server 2000
- \* Hows and Whys of XML Specifications
- \* e-Business Initiatives and ebXML Infrastructure
- \* 4th Generation Application Development with Java
- \* XML-centric Architectures for the Web
- \* Applying XML to Web Collection Indices
- \* Schemas: Rules for XML Documents
- \* Real-Time eBusiness for Competitive Advantage
- \* Sequential XML Processing and XML Standards
- \* Introduction to RDF
- \* Repurpose Your Data: Introduction to XSL
- \* Using Statistics for Structure Mapping



### XML-Journal Sets the Standard

This eagerly anticipated and widely coveted magazine, packed with information written by leading XML gurus, is the only resource you need to understand and make the most of the latest XML advancements.

*Every delegate receives a free one-year subscription to XML-Journal and a one-year **FREE** subscription to Java Developer's Journal - a \$99 value!*

**WWW.XMLDEVCON2000.COM or INFO@CAMELOT-COM.COM**

Presented by: **SYS-CON MEDIA**  
**XML JOURNAL**

Premier  
Sponsor:



Corporate  
Sponsors:



Cosponsor:



Produced by:





placing this type of processing code here as some of the processing options are restricted. It's just as easy to use CFINCLUDE and provide the code at the beginning of those templates that require this type of checking. Putting the code in APPLICATION.CFM would enable the code on every page and that may not be desirable.

The first script in the application, generally INDEX.CFM or DEFAULT.CFM, needs to set the COOKIE and SESSION variables before calling the first script. Unless these variables are set initially, the code will always show the error template.

### Alternatives

There are probably some fairly clever alternatives to the solution presented. Session variables could be updated on each page and CFLOCK used around the entire script. This would effectively lock out the second request from accessing the session variable and would force the submission to wait. This is a brute force method that would require that some error trapping be added to trap a timeout on the CFLOCK.

These types of solutions would have to be used with some caution. Allaire has made significant changes in the CFLOCK tag between versions. Code that works in one environment may fail miserably in

another. Only when the environment is known should such solutions be considered.

CFTRANSACTION has been tried around the code that updates the database. This doesn't work, as the SELECT that checks for the existence of the record will still process. CFTRANSACTION won't stop it since the data doesn't exist until the CFTRANSACTION tag, the SELECT, processes normally.


Consideration could also be given to having some sort of control record in the database that would be updated immediately as the first item in the script. This would, of course, be wrapped in a CFTRANSACTION tag. This would "lock" the record until the update was finished. Unfortunately, it would lock out all other users of the application that may be doing other updates.

Probably the most desperate option is to set up the database connection for only one simultaneous connection. While this is probably a good thing for MSACCESS, other enterprise databases, such as Oracle, would suffer from this restriction.

### The Real Fix

The most desirable fix, in my opinion, is for Allaire to modify the multithreading in the ColdFusion server. What Allaire has done with multithreading works very well

– too well, in fact. There needs to be another check in the ColdFusion server that will simply queue a request with a CFID and CFTOKEN that matches a request that's already running. A developer shouldn't have to make allowances for multiple submissions coming in rapid succession from the same client. There's no logical reason for two requests from the same client to run at the same time.

For many sites this problem may not be an issue as very little database activity is accomplished. But the problem will surface in those sites that run some really hairy queries that take several seconds. Those mysterious duplicate records can probably be traced back to the multithreading provided by the ColdFusion server. A good thing can sometimes cause some undesirable side effects. Knowing what this side effect is, what causes it and ways to guard against it can improve the reliability of the Web site application. 

#### About the Author

Ray Thompson is employed by Q Systems in Oak Ridge, Tennessee, a leading developer of Web-based applications. Ray has over 30 years' experience in information systems and has been involved in many aspects of systems design, deployment and management ranging from "big iron" to desktops.

rayt@qsystems.net

# Sitehosting.net

[www.sighthosting.net](http://www.sighthosting.net)



# Get a Grip on the Clickstream

A brave new world of opportunity

BY  
LEE  
SURMA



Every time users visit your Web site they leave behind footprints of what they did. Usually this clickstream data ends up on your Web server's log.

Many Web site operators, especially those doing e-commerce and B2B, find this information valuable.

To take advantage of this data you can use third-party software or, in some cases, it may make sense to track it yourself. If you wish to do advanced work, such as real-time logging, personalization, real-time customer service and integration with customer data, writing your own custom logging system can get

processed, it looks in the directory tree until it finds an application.cfm file, then runs the code in it. Due to this hierarchical structure, you can isolate your logging to certain areas of your Web site. You may want to start out logging a test area to see how your Web server configuration handles the load and to allow you to work out the kinks before you scale up.

In the application.cfm file you can create code that grabs CGI variables, such as the Web page, IP number and cookie data, and insert them into a database. First set up a file in your database. Hopefully your database is fairly robust, like SQL7 or Oracle. Keep in mind it'll be creating a record for each page view. In a high-volume site you could be creating tens of thousands of records each day, which can present a whole set of maintenance-type problems. Some of the concepts I'm introducing here may be more suited to low volume B2B and e-commerce sites.

## The Code

To get started at a basic level, set up a table in your database that includes the following fields: ID, Date, URL and IP. Name them whatever you like, but follow your naming conventions and be careful about using reserved words (don't use the descriptive names I'm using). The ID field should be a numeric identity. The URL and IP fields should be a string or text type, keeping in mind that this could become a very large file. I use varchar in SQL7.

At this point pick an area of your Web site where you'd like to

start logging and create an application.cfm file. For the beginners out there, there's no magic here. The application.cfm file is a file like any other, but it's treated differently by the ColdFusion server because it's named application.cfm. The code put in this file will run before the code in any other file in a given directory structure.

Use CGI variables returned from the Web server to pick up the Web page and visitor's IP number. A lot of information is available from CGI variables. Turn on debugging for your IP number in the Server Administrator to get a full listing of CGI variables to display after every local page load. I combine the server name, the script name and the query string to come up with a complete URL address. The following code combines them all.

```
<!-- Get ODBC date -->
<cfset DDate=#createodbcdate(now())#>

<!-- Get Page URL -->
<cfif #CGI.query_string# is "">
<cfset
Page="HTTP://"&#CGI.Server_Name#&#
CGI.Script_Name#>
<cfelse>
<cfset
Page="HTTP://"&#CGI.Server_Name#&#
CGI.Script_Name#&"?"&#CGI.query_string#>
</cfif>

<!-- Get visitor IP -->
<cfset IIP=#CGI.Remote_Addr#>
```

At this point insert the variables into the database.

```
<cfquery name="InLog"
```



you moving in the right direction. This article demonstrates one way of doing custom logging and points out some fun things you can do with it.

The key to custom logging with ColdFusion is using the application.cfm file. As you may know, you can put an application.cfm file in any folder within your Web server directory tree. When a .cfm file is

```

datasource="MyDatabase"
dbtype="ODBC">
INSERT INTO Log (Date, URL, IP)
Values (
<CFIF Trim(DDate)is
"">Null<cfelse>#DDate#</cfif>,
<CFIF Trim(Page)is
"">Null<cfelse>'#Page#</cfif>,
<CFIF Trim(IIP)is
"">Null<cfelse>'#IIP#</cfif>)
</cfquery>

```

The code above represents the easy way to do this for presentation purposes. If you're serious about speed and scaling issues, use Stored Procedures and, in some cases, Triggers and Scheduled Execution.

If the Web site contains files that have the .html suffix, change the mapping so these files will be processed by the ColdFusion Server. To do this in IIS, open the Internet Service Manager and go to the properties of the Web site. Once there, click on the Home Directory tab. In the Application Settings area click on Configuration. Create a new mapping for .html with the same executable path as .cfm.

There's a slight performance penalty for doing this since every .html file will now be processed by ColdFusion.

### Big Brother Is Watching You

As people enter your site, you now have real-time clickstream data available. Creating live reports for marketing, management and Webmasters is a simple task.

Now you can start working with personalization techniques. If you have a B2B site in which customers log in with a name and password and/or have an identifying cookie set, you can send e-mail to the salespeople in charge of the account and allow them to view the user's click path. You can also have mechanisms on certain pages that will alert staff to the page hit, telling them who the users are and where they came from. Other more complex applications include having customer service staff watch every unique visit on an e-commerce site. If your technology can handle it, they can attempt to chat with the customer in an effort to obtain a

higher level of customer service. You can also change links and add banner ads and other graphics dynamically, based on a customer's prior click path, in an attempt to enhance his or her visit and present customized sales opportunities.

Some of these things don't necessarily require direct access to clickstream data, but once you implement a custom logging scheme, opportunities for building artificial intelligence into your Web site become more apparent. Most of these require adding conditional logic to the application.cfm file or using Triggers on the database side and sometimes integrating existing customer data.

I've provided a simple example of how to get started with custom logging and some big picture ideas on where it can take you. As you can see, having full control of your Web site's clickstream can open up a brave new world of opportunity.



### ABOUT THE AUTHOR

*Lee Surma is a senior manager of Internet technologies at Public Radio International, a Minneapolis-based distributor of public radio programming. He has five years of experience in various Web-database initiatives.*

LSURMA@PRI.ORG

# Mistral Design Group

[www.purplehosting.com](http://www.purplehosting.com)

# What's FTP?

BY  
KEVIN  
SCHMIDT



When I told them there was a better way to send us large files. They'd been accustomed to sending large image files via an e-mail attachment; I wanted to get them away from this practice and suggested that they use FTP (file transfer protocol). That's where I hit the roadblock. Many of our clients only knew how to send e-mail, surf the Web or create a document in Microsoft Word.

That's the response I got from most of our clients . . .

So rather than spend several hours trying to teach each of them how to use an FTP program, I turned to ColdFusion for a quick and easy solution. With ColdFusion I could create a simple and easy-to-understand Web interface that our clients could use without a fearsome learning curve.

## Getting Started

Two main tags are essential to the implementation of the FTP application: `<CFFILE>` and `<CFFTP>`. The former will be used to transport the files from the user's browser to the ColdFusion server; the latter to transfer the files from the ColdFusion server to the FTP server.

You need to be aware of a few issues when using these tags, though. The major one is security. Giving someone the ability to upload files to your server can be very dangerous, so make sure that only those who are supposed to have access to the application can do this. I enable this application on an as-needed basis only, and I don't recommend leaving it running all the time unless absolutely necessary.

Another issue is that basic ColdFusion security settings may not allow `CFFTP` to execute. To resolve this problem, enable `CFObject` on the basic security page. Now we're ready to examine the steps necessary to implement this FTP application.

## Logging In and Choosing Your File

The first step is to have the client log in. I have the correct login information stored in an Access database, which I use to authenticate the users as well as to track the files they upload. You can use any method you prefer to manage client state; I chose

client variables. I store the users' login information in a client variable for use when they begin to transfer files.

After the login, the next step is to select the file they want to upload via a standard HTML form. The following code shows the basic structure for setting up the form.

```
<FORM ACTION="sendfile.cfm"
METHOD="post"
ENCTYPE="multipart/form-data">
<INPUT TYPE="file" NAME="filetosend"
SIZE="30">
<INPUT TYPE="submit" NAME="SEND">
</FORM>
```

The main difference between this form and a regular form is the `ENCTYPE="multipart/form-data"`, which is necessary to process the uploaded file. After the appropriate file has been selected to send to the server and the send button has been clicked, the file transfer process starts.

## Introducing <CFFILE>

Once the process has been started, the `<CFFILE>` tag will transfer the file to a folder specified on the ColdFusion server. Three attributes of the tag are required to upload a file, plus several that are optional. First is the `ACTION` attribute, which needs to be set to `UPLOAD`. Also required is the `DESTINATION` attribute, which specifies the location where the file is to be saved on the ColdFusion server. The full path to the directory needs to be listed, and the trailing slash also needs to be included. The final parameter is `FORMFIELD`, which is the name of the form field that the users specified their file in. Note that it isn't necessary to include the # signs with this field.

The optional attributes for `<CFFILE>` can be very useful. The first, `NAMECONFLICT`, tells the server what to do if a file with the same name already exists on the server. With this attribute you can tell the server to `ERROR` without saving the file; `SKIP`, which neither saves the file nor throws an error; `OVERWRITE`, which will overwrite the file; or `MAKEUNIQUE`, which tells the server to create a unique file name.

The next attribute, `ACCEPT`, allows you to specify what types of files are accepted. For example, if you want to allow only GIFs or JPGs to be uploaded, you can set the attribute to:

```
ACCEPT="image/gif", "image/jpg"
```

This attribute also helps with security concerns if you know that only certain types of files are going to be uploaded.

The next attribute, `MODE`, which applies only to UNIX users, allows you to specify permissions. Thus `MODE=777` assigns the file read, write and execute permissions for everyone.

The final attribute, `ATTRIBUTES`, allows you to set the following attributes on a file being uploaded: `ReadOnly`, `Temporary`, `Archive`, `Hidden`, `System` and `Normal`. If the attribute isn't used, the file maintains its original attributes.

```
Sample Code for <CFFILE>
<CFFILE ACTION="Upload"
FORMFIELD="filetosend"
DESTINATION="C:\cffile\uploads\"
NAMECONFLICT="MakeUnique"
ACCEPT="image/gif, image/jpg"
>
```



# Allaire

**[www.vue.com/allaire](http://www.vue.com/allaire)**

Once the file has been uploaded, a number of parameters are available to check the status of the upload. You can check the file name and directory in which the file was saved; whether or not the file was actually saved; how big the file is; and, most important, the actual file name that was saved to the server. This variable will be used to continue the process and actually move the file to its final destination on the FTP Server. I've included the full list of attributes and

file on the FTP server. This is where the `#ServerFile#` variable comes into play from the `<CFFILE>` status variables. `#ServerFile#` is the name of the file that it was given on the ColdFusion server. This will supply the correct file name in the event that it was changed from the original file name on the user's machine. This would happen if the `MAKEUNIQUE` attribute was specified and the file already existed on the ColdFusion server. Use this variable to specify

the number of retries until a failure is reported. The default is 1. `STOPONERROR` allows you to specify whether to stop on an error. If yes, all processing stops and an error message is displayed. If no, the error information is saved to three variables: `CFFTP.Succeeded` (yes or no), `CFFTP.ErrorCode` (Error Number) and `CFFTP.ErrorText` (Text explaining error type). `PASSIVE` is a boolean value indicating whether to enable passive mode. The default is disabled.

### File Upload Parameters for <CFFILE>

*These are the parameters, word for word, from the CF 4.5 CFML Language Reference published by Allaire.*

**AttemptedServerFile:** Initial name ColdFusion used attempting to save a file, for example, myfile.txt

**ClientDirectory:** Directory location of the file uploaded from the client's system

**ClientFile:** Extension of the uploaded file on the client's system without a period, for example, txt not .txt

**ClientFileName:** Filename without an extension of the uploaded file on the client's system

**ContentSubType:** MIME content subtype of the saved file

**ContentType:** MIME content type of the saved file

**DateLastAccessed:** Date and time the uploaded file was last accessed

**FileExisted:** Indicates (Yes or No) whether or not the file already existed with the same path

**FileSize:** Size of the uploaded file

**FileWasAppended:** Indicates (Yes or No) whether or not ColdFusion appended the uploaded file to an existing file

**FileWasOverwritten:** Indicates (Yes or No) whether or not ColdFusion overwrote a file

**FileWasRenamed:** Indicates (Yes or No) whether or not the uploaded file was renamed to avoid a name conflict

**FileWasSaved:** Indicates (Yes or No) whether or not ColdFusion saved a file

**OldFileSize:** Size of the file that was overwritten in the file upload information

**ServerDirectory:** Directory of the file actually saved on the server

**ServerFile:** Filename of the file actually saved on the server

**ServerFileExt:** Extension of the uploaded file on the server, without a period, for example, txt not .txt

**ServerFileName:** Filename, without an extension, of the uploaded file on the server

**TimeCreated:** Time the uploaded file was created

**TimeLastModified:** Date and time of the last modification to the uploaded file

their definitions from the CFML language reference in the sidebar.

### <CFFTP>

This brings us to the final step in the process – moving the file from the ColdFusion server to the FTP server and deleting it from the ColdFusion server.

The `<CFFTP>`, like the `<CFFILE>` tag, has required and optional attributes. The tags required to upload to the server are `ACTION`, `SERVER`, `USERNAME`, `PASSWORD`, `LOCALFILE` and `REMOTEFILE`. For uploading, the `ACTION` attribute must be set to `PutFile`. The `SERVER` attribute is just what it sounds like – the server name, that is, `ftp.yourdomain.com`. `USERNAME` and `PASSWORD` are also pretty self-explanatory. This is where I use the client variables that I saved from the earlier login. `LOCALFILE` is the name of the file on the ColdFusion server, and `REMOTEFILE` is the name of the

both the `LOCALFILE` and `REMOTEFILE` attributes.

The optional attributes for `<CFFTP>` tag supply an array of useful tools. `TIMEOUT` allows you to specify the length in seconds for the timeout of all operations. The default is 30 seconds. `PORT` allows you to specify a port value for the FTP server if different from the default of 21. `CONNECTION` is used to name your connection if you want to cache it. If you make a call to `<CFFTP>` with the name attribute specified, it will use the same FTP information as the other connection of the same name, if it was cached. `PROXYSERVER` allows you to specify a proxyserver to use if proxy access was specified. If not, the tag retrieves proxy information from the registry. `PROXYBYPASS` allows you to specify a local IP and bypass the proxy server. If this attribute isn't specified, the tag reads the bypass list from the registry. `RETRYCOUNT` allows you to specify

Sample Code for <CFFTP>

```
<CFFTP
    ACTION="PutFile"
    SERVER="ftp.yourdomain.com"
    USERNAME="#client.username#"
    PASSWORD="#client.password#"
    LOCALFILE="#serverfile#"
    REMOTEFILE="#serverfile#"
>
```


### Final Step – File Removal

The last step in the process is to remove the file from the ColdFusion server, using the `<CFFILE>` tag with the `ACTION` attribute set to `DELETE`. The file to be deleted must be specified using the `FILE` attribute. Once again, you can use the `#ServerFile#` variable from the `<CFFILE>` status variables. When this is finished, the file will be on the FTP server and will also have been removed from the ColdFusion server.

Sample Code for <CFFILE>

```
<CFFILE ACTION="Delete"
    FILE="C:\cfile\uploads\#ServerFile#"
>
```

### Summary

ColdFusion offers an impressive variety of tools for file manipulation and FTP. This example touches on only one facet of many that can be implemented using these powerful tags. I've discovered that most of our clients find this an easy and understandable way to send us their files. It's simplified their lives and made my job a little less hectic. I'm sure the time I saved by using this solution (versus training our clients on FTP) should help me take that vacation I've been planning. 

SCHMIDT@PWB.COM

ABOUT THE  
AUTHOR  
Kevin Schmidt is  
Internet services  
manager for PWB,  
an integrated  
marketing and  
information design  
firm in Ann Arbor,  
Michigan. He spends  
most of his time  
developing Web  
applications for clients  
using ColdFusion  
and Oracle.

# **Developers Network**

**[www.developersnetwork.com](http://www.developersnetwork.com)**

# Customize ColdFusion Authentication

## Utilizing your user database

BY  
SARGE

After reading several ColdFusion Forum (<http://forums.allaire.com>) postings and receiving numerous e-mails detailing developers' pains in setting up Advanced Security in ColdFusion so as to implement page-level user security with the <CFAUTHENTICATE> paradigm, I decided to write an article describing an alternative methodology for authentication in a CF application.

In fact, I know you've been expecting this information from me.

My focus is user authentication and access control within a given CF application at the page level. Development and deployment concerns, such as locking down developer resources – CF tags and functions, data sources, implementing source control and security sandboxes – are handled nicely within CF's Advanced Security, but they go beyond the scope of this article. Also, access controls can be created within Web servers to lock down user access to directories containing code based on the authenticated user's credentials.

### Authentication Problem

Suppose you want to display or hide some functionality within any given page in your CF application based on a user's privileges (group memberships) without querying a user database (e.g., LDAP, ODBC) on every request. This could be something as benign as hiding a link to the Manager's section of your site or changing the section graphic based on a user's organization membership. How do you implement this type of functionality without CF's Advanced Security?

One solution is to populate a session-level structure with a user's group memberships (or access levels) when the user authenticates to the site. As far as authentication paradigms are concerned, choose your poison. You decide if you're going to apply username/password combinations or X.509 user certificates. For this article I'm using an LDAP user directory (Netscape's Directory Server 3.x), CF Enterprise

4.5.1, Netscape's Enterprise Server 3.x and domestic strength X.509 user certificates on my laptop. This solution has also been utilized in a production environment with CF 4.01 Professional and IIS, using an ODBC user directory with a username/password combination. (If you want to use the NT SAM, you'll have to implement third-party solutions, like the CFX custom tags in the Developer Exchange [[www.allaire.com/developer/gallery/index.cfm?objectID=6800](http://www.allaire.com/developer/gallery/index.cfm?objectID=6800)]. The meat of this functionality is all done in my application's Application.cfm file, which can be downloaded from the **CFDJ** Web site.

### Authenticating the User

First, create the Application framework with a <CFAPPLICATION> tag. Don't forget to specify the SESSIONMANAGEMENT and SESSIONTIMEOUT parameters (see Listing 1) To instantiate this session state, I've molded several techniques of creating and maintaining session and client management that I've seen in a number of CF Forum and CF\_Talk ([www.cfadvisor.com/cft\\_frame.htm](http://www.cfadvisor.com/cft_frame.htm)) postings. I'm not including the details of the session and client management code in my listings, but this management mechanism will work in clustered environments and with cookies disabled in the user's browser.

Next, query your user directory for the authenticated user's credentials. The cn attribute of X.509 certificates maps directly to corresponding entries in my LDAP. Two queries to the LDAP are necessary to grab the user's groups with a high

degree of certainty – I'll explain in a minute. CF 4.x exposes several CGI environment variables extracted from a user's certificate: AUTH\_USER, AUTH\_TYPE, CLIENT\_CERT, CLIENT\_CERT\_ISSUER\_DN, CLIENT\_CERT\_SSL\_ID, CLIENT\_CERT\_USER\_DN. (Although I don't use these variables in this code, knowing them will prove useful for certificate management – but that's another topic.) Another variable, AUTH\_GROUP, should contain the authenticated user's group memberships, but I've never seen it populated. Once the user authenticates, the AUTH\_USER CGI variable is used as a filter in a <CFLDAP> query to the LDAP, specifying useful attributes to retrieve – for example, DN, given-Name, SN, telephone number, pager and mail (see Listing 2).

Now loop through the record set returned from the <CFLDAP> query and create a structure container in the SESSION scope. Specifying useful fields in your <CFLDAP> query creates a container that serves as a nice user profile that's cached in server memory during the user's session. A complete Distinguished Name (DN) attribute is needed to correctly retrieve all the user's group memberships. Use the DN attribute from the first <CFLDAP> query as the DN attribute for a second <CFLDAP> query, which will retrieve the user's groups (see Listing 3).

Finally, extend your SESSION.siteUser structure and add the retrieved memberships in a comma-delimited list. Again, this second <CFLDAP> query retrieves all group memberships based on the authenticated user's full DN. The DN at-

# Allaire Sandcastle

**[www.allaire.com](http://www.allaire.com)**

tribute ensures the user's uniqueness in the LDAP (see Listing 4).

Now that the authenticated user's profile information and group memberships are stored in his or her session, using that information is simple if you know how to work with structures. You can easily extrapolate the user information you need by looping through the `SESSION.siteUser` structure (see Listing 5).

### Art of Authentication

Implement access control within your templates by wrapping objects, code – even the entire template itself – in an IF-Else block that inspects the `Session.siteUser.Group` key. For example, you may want to programmatically display links and/or text based on an authenticated group. Or you may just want to dynamically display a custom group header on individual pages within a site. Simply control the value in the source attribute of your image tag with some form of an IF statement. However, the complexity of your IF-ELSE statement or block will depend on your user database structure and group memberships. This same design can be used to wrap CF tags, functions and variables within your code to programmatically provide a self-enforced security framework outside Advanced Security (see Listing 6).

### Notes

#### Access Control

Most Web servers provide site and directory-level access control,

requiring user authentication to access folders or directories containing the Web site files. I'm not privy to native methods of extending this functionality beyond the template and/or page level to the elements within the code – for example, links, images, text or variables. This custom design does extend the Web server's own access control to the element level via CF, using the same user database for authentication.

#### Authentication Functions


ColdFusion provides tags and functions (e.g., `IsAuthenticated` and `IsAuthorized`) that coincide nicely with the Advanced Security paradigm, but they aren't available to you under this customized solution. ColdFusion's Advanced Security is designed to provide scalable, granular security for application development and deployment to ISPs and managed development environments. While the customized security scheme described in this article provides scalable, granular access control to template elements within an application and, to some degree, overall application security, it's recommended as an alternative solution only for access control, not as a substitute for Advanced Security.

#### Performance Hits

If you're concerned about the performance hit suffered by the queries to the user database in the `Application.cfm`, don't worry. True, the queries potentially run with every template call due to the very

nature of the `Application.cfm` template. However, various session management schemes and coding methodologies are available to eliminate this issue. The coded session scheme presented here is just one method of ensuring that the user is authenticated only once per site visit. Again, session management isn't discussed here, but this method is clean and doesn't rely on cookies.

#### User Database/Directories

For the purposes of this article I used an LDAP user directory, but you can substitute any user directory accessible to CF – ODBC, LDAP (NDS, Oracle, etc.) or NT SAM (Active Directory). I also used X.509 client certificates within my authentication scheme, but the solution can be worked with username/password combinations. The key to the authentication is creating and capturing an attribute that uniquely identifies the user once he or she authenticates against your user database. For LDAP this is the user's full DN; for NT SAM it would be the user's SID. For ODBC this uniqueness must be coded within table schema (Auto-Identity, Composite Primary Key, etc.). Regardless of how this uniqueness is defined, it's tantamount to the process described above and will ensure both customized session management and security. 

SSARGENT@ALLAIRE.COM

#### ABOUT THE AUTHOR

Sarge is a senior consultant with Allaire Corporation. He has been coding advanced applications in CF since version 2.0.

#### Listing 1

```
<cfapplication name="UserAuthApp"
    clientmanagement="Yes"
    sessionmanagement="Yes"
    setclientcookies="Yes"
    sessiontimeout="#CreateTimeSpan(0, 0, 20, 0)"#
    applicationtimeout="#CreateTimeSpan(1, 0, 0, 0)"#
    clientstorage="Cookie"
    setdomaincookies="Yes">
```

```
<cfset SESSION.newUserYN = IsDefined("SESSION.userAppInit")>
```

```
<CFIF SESSION.newUserYN is "No">
```

#### Listing 2

```
<cfdap action="QUERY"
    name="getUser">
```

```
    attributes="dn,sn,givenname,ou,uid,cn,telephonenumber,pager,mobile,mail"
    start="ou=Consulting, ou=Support Services, o=Allaire, c=US"
    scope="SUBTREE"
    filter="cn=#CGI.AUTH_USER#"
    server="sargesldap.allaire.com"
    username="#ldapusername#"
    password="#ldappassword">
```

#### Listing 3

```
<cfscript>
    SESSION.SiteUser = StructNew();
    tmp = StructInsert(SESSION.SiteUser, 'Name', getUser.given-
    Name & ' ' & getUser.sn, True);
    attribs = getUser.ColumnList;
    for (i=1; i LTE ListLen(getUser.ColumnList); i = i+1) {
        col = ListGetAt(attribs,i,','); val = "getUser."&#col#;
        tmp = StructInsert(SESSION.SiteUser, col, evaluate(val),
        True);
```



### Listing 4

```
<cfldap action="QUERY"
name="getGroups"
  attributes="cn,owner"
  dn="#SESSION.SiteUser.DN#"
  start="#Application.groupbase#"
  scope="SUBTREE"

filter="(&(objectclass=groupofuniquenames)(uniquemember=#SESSION.SiteUser.DN#))"
  server="sargesldap.allaire.com"
  username="#ldapadmin#"
  password="#ldappassword#">

<cfscript>
  tmp = StructInsert(SESSION.SiteUser, 'Groups',
valuelist(GetGroups.cn));
</cfscript>
```

### Listing 5

```
<table border='1'>
<cfloop collection="#Session.SiteUser#" item="myItem">
<cfset myAttrib = "Session.SiteUser."&#myItem#>
<cfoutput>
  <tr>
    <td>#myItem#:</td>
    <td>#evaluate(myAttrib)#</td>
  </tr>
</cfoutput>
</cfloop>
</table>
```

### Listing 6

```
<CFSCRIPT>
IF (ListFindNoCase(Session.siteUser.Group, "Managers", ","))
```

```
{
  IsMgr = "1";
} ELSE {
  IsMgr = "";
}
</CFSCRIPT>
```

Links:

```
<A href="profile.cfm">Edit Profile</A><BR>
<A href="schedule.cfm">Schedule</A><BR>
<A href="timecards.cfm">Time Cards</A><BR>
<CFIF LEN(TRIM(isMgr))>
  <A href="bonus.cfm">Promotions</A><BR>
  <A href="walknpapers.cfm">Pink Slips</A><BR>
</CFIF>
<A href="Expense.cfm">Expense Reports</A>
```

Image Header:

```
<CFOUTPUT>

</CFOUTPUT>
```

CF Variables & Tags:

```
<CFIF LEN(TRIM(IsMgr))>
<CFSET DataSource = "MgrDB">
<CFINCLUDE Template="hirefire.cfm" AddToken= "No">
</CFIF>
```

```
<CFQUERY Name="Generic" Datasource="#DataSource#">
  SQL Statements . . .
</CFQUERY>
```

CODE  
LISTING

The code listing for  
this article can also be located at  
[www.ColdFusionJournal.com](http://www.ColdFusionJournal.com)

# Woodbourne Solutions

**www.woodbournesolutions.com**

# Woodbourne Solutions

**www.woodbournesolutions.com**



# LEVERAGING BUSINESS IN THE UNWIRED WORLD

Register by  
October 27 and  
**SAVE  
\$200**

## Learn from

Wireless DevCon 2000 will be a unique combination of superior technical sessions and an interactive show floor—designed to introduce the latest world-wide wireless applications to Silicon Valley's development community. You'll be joined by fellow wireless enthusiasts—the

## CONFERENCE:

December 3–5, 2000

DoubleTree San Jose  
San Jose, CA

## EXHIBIT:

December 4–5, 2000

Plan to be a part of the only  
vendor-neutral conference  
and expo designed specifically  
for the wireless software  
developer.

[www.wirelessdevcon2000.com](http://www.wirelessdevcon2000.com)

Presented by:



Media Sponsors:



Produced by:



Official News Distributor:





## EVENT HIGHLIGHTS

- Independent, vendor-neutral technical program—3 tracks, 40+ sessions
- The best of the latest wireless software developments and initiatives from Europe and around the world will be presented by recognized industry experts.
- Night School Sessions—Extend your day program or take independently. Select from 6 in-depth topics on Sunday and Monday nights.
- Two-day Interactive Exhibition on Monday and Tuesday.
- Take advantage of this independent, vendor-neutral presentation of the industry's most technically advanced wireless event. Walk away with the programming skills and the technical confidence you need to ensure you are maximizing the full potential that wireless technology can provide your enterprise application.
- Gold Passport allows you to take advantage of all sessions most economically.

## WIRELESS DEVCON 2000 IS DESIGNED FOR...

- Programmers, developers, engineers
- Software architects
- System engineers
- Web developers
- Product managers
- Project leaders
- Consultants
- Educators

## TOPICS INCLUDE:

- Developing for the Entire World: Internationalization and Single Standards
- 3G: Mobile Multimedia, Multimedia Messaging, Electronic Wallets, Worldwide Compatibility and the Convergence of Digitalization
- Open-Source Gateways and Open-Source Software that Support Wireless Application Development.
- Getting Started with Wireless Development
- Common-Use Cases and the Software to Support Them
- Using Device Capabilities and User Profiles (CC/PP) to Customize Content.
- Using Localization Information to Customize Content
- Technologies and Architectures for Content Customization: XSLT, JSP, ASP; Custom Transcoders
- Design Issues in Repurposing (Transcoding) Web Content for Mobile Devices
- Security Issues and Techniques
- Multi-modal Interactions: Point and Tell, Speak and Click
- Java and JTAPI: The Emerging Platform for CTI, Media Services and IP Telephony
- Integrating Mobile Clients with Enterprise Architectures
- Portal Strategies for Mobile Operators
- WAP and Its Alternatives and Competitors: What Are the Best Developer Strategies?

## TWO-DAY EXHIBIT

A full-scale exhibit hall will be packed with leading vendors who will demonstrate the latest products and answer your questions. All are welcome to fun-filled networking opportunities.

## SPECIAL EVENTS INCLUDE:

- Welcome reception sponsored by *Wireless-Journal*
- Keynotes by industry leaders
- Management briefings
- Vendor presentations
- Birds-of-a-Feather sessions

## TECHNICAL CHAIRS

Alexander Nakhimovsky, Ph.D., has been teaching computer science at Colgate University since 1985. He is the coauthor of *JavaScript Objects*, *Professional Java XML Programming* and *Professional Java Programming for the Wireless Web*.

Tom Myers, Ph.D., is a software developer and consultant, working most recently on a Java-XML infrastructure for Web and wireless applications. He is co-author of *JavaScript Objects*, *Professional Java XML Programming*, and *Professional Java Programming for the Wireless Web*.

## INTRODUCING SYS-CON MEDIA'S WIRELESS-JOURNAL

Come meet the writers and editors of *Wireless-Journal*. Exchange ideas with fellow wireless developers. Share your programming experiences and network with the industry's wireless leaders.



## TUNE INTO SYS-CON RADIO FOR LIVE COVERAGE OF WIRELESS DEVCON 2000

SYS-CON Radio will bring this event LIVE at [WirelessDevCon2000.com](http://WirelessDevCon2000.com) to wireless developers around the world. Don't miss live interviews, product announcements and discussions with the movers and shakers of the wireless industry.



To exhibit contact Jeff Adler at [jadler@camelot-com.com](mailto:jadler@camelot-com.com). For speaking opportunities, contact Stewart Quealy at [stewart@camelot-com.com](mailto:stewart@camelot-com.com)

the Experts...  
industry's most respected  
technical experts, sought-after  
gurus and advanced users.  
Benefit from their expertise.  
Immerse yourself in four days  
of wireless techniques and  
master new skills taught by  
those who are defining  
wireless's future.

## BENEFITS OF ATTENDING INCLUDE...

- Learn your options in a vendor-neutral, independent environment.
- Have your questions answered by leading wireless experts.
- The tips and techniques you'll learn will help you do your job better.
- Discover new applications being developed today that you will need tomorrow.
- Sessions are designed for the users at all levels with special sessions just for gurus.
- Network with fellow software developers as well as recognized wireless experts.
- You'll learn how wireless is transforming the Internet and how you build your large-scale enterprise applications.
- The industry's recognized wireless experts have designed the technical program which will include 40+ sessions with over 70+ hours of instruction.

# Another Custom Tag

# DHTML

# Wrapper

An inside look

BY CHRISTIAN SCHNEIDER

This article demonstrates how to write a paired custom tag that encapsulates complex DHTML logic into a simple-to-use tag. I was inspired to write by Tim Buntel's article about DHTML wrappers in *CFDJ* (Vol. 2, issue 4). It motivated me to extend a custom tag I had developed and used throughout my projects.

As you can see from Figures 1 and 2, my custom tag acts like a nice widget that enables Web pages to become user-friendly by providing a mechanism that collapses and expands content, then wraps it with a titled border. It groups Web sites into several sections that a user can expand and/or collapse. This kind of grouping is helpful when used in any kind of report that presents different sections.

## Types of Custom Tags

The objective of ColdFusion's custom tags is to encapsulate complex code into an easy-to-use (and easy-to-reuse) interface.

Basically, you can write three different types of (custom) tags: single, paired and nested parent-child. You've used each type when working with the usual ColdFusion tags:

1. **<cfinclude template="header.cfm">**: A single CF tag that's useful for any kind of action that should be initiated
2. **<cfoutput> some content here </cfoutput>**: The most popular paired CF tag, useful when working on the content included in your custom tag
3. **<cftr> <cftritem value="aSampleValue"> ... </cftr>**: A good example of the nested parent-child variant, useful when working with hierarchical parent-child architectures; displays a hierarchical tree

Since my custom tag operates on content by wrapping it with an expand/collapse layer, it should definitely be developed as a paired custom tag. Basically all custom tags (no matter what type) have the ability to carry attributes, an important factor

# **Fall Internet Wotld**

**[www.pentonevents.com](http://www.pentonevents.com)**



since custom tags should be as reusable as possible. For a custom tag to be reusable, developers must be able to affect and change what it does (for custom tags that encapsulate visual effects). But developers don't like using a lot of different attributes in a simple situation. To provide easy-to-use custom tags, consider making their attributes optional by providing default values for them when they're not used.

For this article's custom tag that wraps its content with a titled expand/collapse border, I made up a list of attributes that enables potential users to have as much control as possible. The following attributes are optional:

- **Border="yes/no"**: Defines whether or not a visual border should be drawn (default: YES)
- **BorderSize="size in px"**: Sets the size of the visual border in pixels (default: 3)
- **BorderColor="color"**: Sets the color of the visual border to a named color or an #RGB value (default: lightgray)
- **Image="image"**: Sets the image to use as the click-control
- **ImageOpen="image"**: Sets the image to use when expanded
- **Title="string"**: Sets the border's title
- **TitleClosed="string"**: Sets a title to use when the layer is collapsed
- **TitleTooltip="string"**: Sets the tooltip for the border's title
- **TitleFont="font"**: Sets the font of the border's title (default: Arial)
- **TitleColor="color"**: Sets the color of the border's title (default: the page's default text color)
- **TitleColorOpen="color"**: Sets the color of the border's title when the layer is expanded (default: the same title color you set previously)
- **TitleSize="size in pt"**: Sets the size of the border's title (default: 10)
- **IsClosed="yes/no"**: Defines whether the layer should be initially collapsed or expanded (default: no)
- **Width="width in px or %"**: Sets the width of the layer as an absolute pixel value or as a percentage (default: 100%)

I've shown what this custom tag is intended for and what attributes it can take. Here's an example of how to use it:

```
<cf_openLayer title="My Test" bordercolor="Red">
```

This content can be made of anything... Put HTML, CFML, Applets, Images and even other nested instances of this custom tag here...

```
</cf_openLayer>
```

This simple example generates a red expand/collapse border titled "My Test" around the content. All other attributes not specified here are initialized with their default values inside the custom tag's code.

## Now for the Dirty Part

So far I've demonstrated what the custom tag does and how to use it. This is enough for developers who don't like to get in touch with the code inside the custom tag but merely want to use the effect on their pages. This encapsulation is the main reason custom tags are so widespread and have gained such broad acceptance in the developer community – ease of use without concern for the inside details.

For those who'd like to go inside, I'll explain briefly how the dirty part – the DHTML code to generate the effect of expanding/collapsing content – was done. In Internet Explorer the visual effect of expanding and collapsing content using DHTML is made possible by the DOM (Document Object Model), which makes every HTML tag with its styles and properties transpar-

ently accessible to the developer. I can easily change the style of the layer holding the content from expanded to collapsed and vice versa. In Netscape, which doesn't have a DOM, this isn't as easy to achieve. Instead, Netscape makes HTML dynamic by making certain properties of layers accessible. I can change the layers' position and content, and even show and hide them, but I can't collapse them to free the space. Okay, it actually can be done, but with a completely different approach: dynamically rewriting the Web page's HTML code with JavaScript. Since this would break the handy way to incorporate it into a custom tag to use anywhere on a page (even multiple times, and even nested), this article's custom tag is made especially for Internet Explorer. "Hey, what about all the other browsers? If they can't produce the effects, please don't let them crash with JavaScript errors...." As this statement was the first thing that came into my mind, I made it browser-safe using ColdFusion. Using ColdFusion? Yes, that's possible. It's a convenient (and safe) way to make any kind of client-side effects browser-safe; ColdFusion's built-in variable #CGI.HTTP\_USER\_AGENT# sniffs out what browser the client is accessing the current template with. Using this variable, the custom tag checks whether the client is using Internet Explorer 4 (or higher) or any other browser and stores this in a Boolean variable, which is used to dynamically send different code to the client: either the full-blown DHTML and JavaScript code, or – if accessed with another browser – simple HTML code.

As mentioned above, all types of custom tags can carry attributes. These attributes (like those in the sample snippet above) are accessible inside the custom tag's code via the ATTRIBUTES variable scope, which holds all attributes and their values like a structure. The two attributes in the sample above can be accessed inside the custom tag using #ATTRIBUTES.title# and #ATTRIBUTES.bordercolor#, respectively. By using <cfparam> you can make attributes optional by providing them with default values as in this statement:

```
<cfparam name="ATTRIBUTES.bordersize" default="3">
```



FIGURE 1: Sample view of the custom tag wrapping some content



# **JAVACON**

**[www.javacon2000.com](http://www.javacon2000.com)**

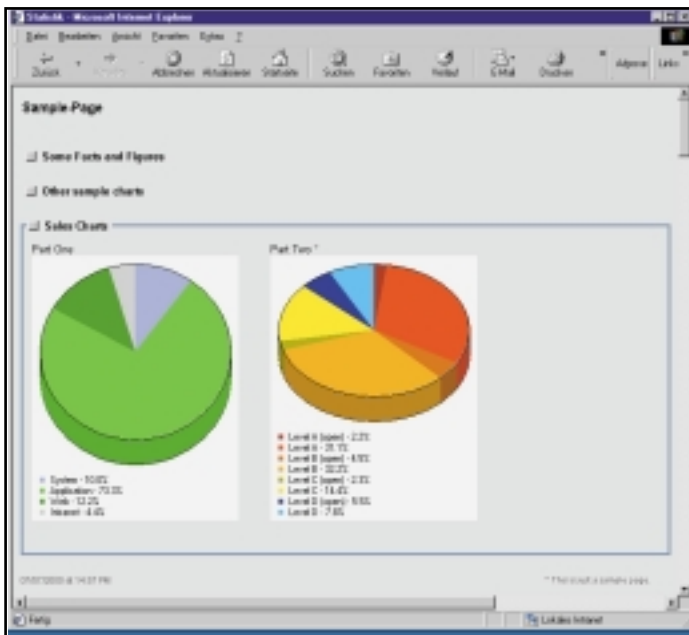


FIGURE 2: View of the custom tag with some content collapsed

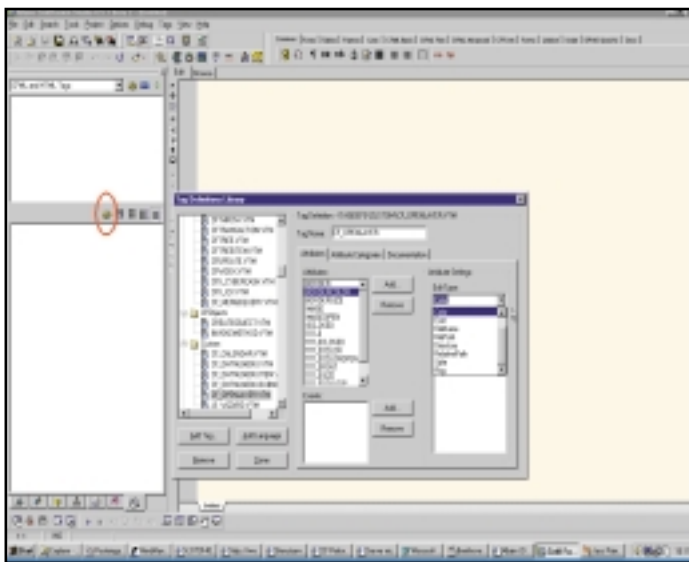


FIGURE 3: Visual VTML editor in ColdFusion Studio

This is used inside the custom tag to set the default border size to three pixels if a size wasn't provided.

By accessing the content that's wrapped by the starting and closing tags, I can make a custom tag into a paired custom tag. To put it simply, all the content that the starting and closing tags wrap inside is stored in the variable `#THIS.TAG.GeneratedContent#`, which can be manipulated depending on what you wish your custom tag to do. `THIS.TAG` is a special structure holding information for custom tags regarding the way they were called. To check whether the paired custom tag is used correctly (meaning it actually has an ending tag), look in the Boolean variable `#THIS.TAG.HasEndTag#` would be true if the custom tag were called with an ending tag provided and false if not. This valuable information should be checked first in all paired custom tags so they can stop executing and throw an exception when the user has forgotten to use the closing tag.

The `THIS.TAG` structure provides you, the developer of custom tags, with interesting information: the string in `#THIS.TAG.ExecutionMode#` is either `START` or `END` depending on whether the starting or closing tag is being processed. Using

this mode flag, you can write paired custom tags that do some work when the starting tag is processed, and more when the closing tag is processed. I coded my custom tag that way: at the processing of the starting tag it opens the titled border as a layer and draws the click-control with which the users can expand and collapse the content. At the processing of the ending tag it simply closes the layer and its border (see Listing 1).

Another interesting thing in this custom tag's code is the JavaScript for expanding and collapsing the layer holding the wrapped content. The effect is achieved by assigning either "none" or "block" as the value of the current layer's "display" style property. What makes it so interesting is that you can use the custom tag as often as you want on one page, because it assigns each layer a randomly generated, unique ID that's used to address each layer separately. The general JavaScript code used to generate the effect is put out on the page only once. This is made possible by storing a special flag in the `REQUEST` variable scope when the JavaScript code is written on the page for the first time. This flag is checked for every subsequent instance of this custom tag on the requested page. As the name might suggest, the `REQUEST` variable scope is available for all files (included files, custom tag, and more) for the current request accessing the page. That way the custom tag can check whether it was already used on the current requested page. If it was, it only needs to draw the layer and doesn't need to write redundant JavaScript code.

I hope my brief explanation of how this custom tag works inside and what techniques were used is sufficient. Unfortunately, showing all the details of the DHTML effect is beyond the scope of this article.

## VTML Please

Wow! Besides HTML, DHTML, CFML, XML and WML, there's still room for another -ML abbreviation: VTML (Visual Tools Markup Language), the great family of markup languages that customize Allaire's ColdFusion Studio. The wonderful thing about VTML is that it's derived from XML, so you can easily get started based on your existing knowledge of tag-based markup languages. Using VTML you can control all the helpful things Allaire made available in ColdFusion Studio – Tag Inspector, Tag Insight, Tag Editing Dialogs, CF Studio Wizards and more. It's helpful for custom tag developers, since it enables them to tightly integrate their tags into the ColdFusion Studio IDE. It's easy to do: simply provide the article's custom tag with a VTML file for ColdFusion Studio's Tag Insight. I'm sure every developer working with Studio is using the Tag Insight feature (enabled from the Settings Dialog) quite often while typing HTML and CFML tags. It's the handy popup box that lists all possible attributes of the tag you're currently typing. In addition to making this kind of integration with Studio possible, Allaire has taken another step forward by providing us with visual tools for the VTML development inside Studio. As you can see from Figure 3, the visual VTML editor for the Tag Insight can be called from the Resource Tab's Tag Inspector (click on the red marked icon). To generate a Tag Insight for your custom tag in the VTML editor:

1. Click on the Add Tag... button and enter the custom tag's name (here `CF_openLayer`).
2. Click the Add... button on the right tab to add attributes to your custom tag.
3. Define the type of each attribute by choosing the appropriate one from the Edit Type drop-down listbox:
  - **Text:** For attributes taking strings as their value
  - **Enumerated:** For attributes taking one of a list of predefined values

# **NET DIVE**

**[www.netdive.com](http://www.netdive.com)**

- **Color:** For attributes representing a color value
  - **Font:** For attributes representing a font definition
  - **FileName:** For attributes holding a file name
  - **FilePath:** For attributes holding a file path
  - **Directory:** For attributes holding a directory name
  - **RelativePath:** For attributes holding a relative path
  - **Style:** For attributes holding a style definition
  - **Flag:** For attributes acting as flags
4. Click the Done button and restart

ColdFusion Studio.

5. Open a new template and start typing `<cf_openLayer>`, then try to set some attributes. Voilà! Tag Insight is working for this custom tag.

What about the VTML? That was done by the VTML editor. If you're curious, Listing 2 provides exactly what the VTML editor has generated. After using the editor, you can also find this file on your workstation in the subdirectories of `C:\Program Files\Allaire\ColdFusion Studio\Extensions\TagDefs\`, depending on where you installed ColdFusion Studio. You can distribute this VTML file along with your cus-

tom tag to enable developers using your tag to seamlessly integrate it into Studio.

While writing this article I submitted this custom tag and its VTML file to Allaire's Tag Gallery ([www.allaire.com/developer/gallery.cfm](http://www.allaire.com/developer/gallery.cfm)), so you can download it from there. Watch for updated and enhanced versions.



#### About the Author

Christian Schneider is a ColdFusion developer with over three years of intensive experience in developing CF-based intranet applications for banks and logistic corporations.

mail@christian-schneider.de

#### Listing 1

```
<!-- openLayer.cfm -->

<!--
*****

<CF_openLayer>
Version 1.0
Wraps its content with an collapse-expand-layer...

(This layer only works in IE4++
To be browser-safe: when accessed with another
browser, only a title is shown without the layer.)

Usage:

    <CF_openLayer
        [ Border="yes|no" ]
        [ BorderSize="size" ]
        [ BorderColor="color" ]
        [ Image="image" ]
        [ ImageOpen="image" ]
        [ Title="string" ]
        [ TitleClosed="string" ]
        [ TitleTooltip="string" ]
        [ TitleFont="font" ]
        [ TitleColor="color" ]
        [ TitleColorOpen="color" ]
        [ TitleSize="size in pt" ]
        [ IsColsed="yes|no" ]
        [ Width="width in px or %" ]&gt;

    ...
</CF_openLayer>

Contact:
mail@Christian-Schneider.de
www.Christian-Schneider.de

*****
-->
<cfsetting enablecfoutputonly="Yes">

<!-- When no Ending-Tag is used throw an Exception -->
<cfif ThisTag.HasEndTag EQ false>
    <CFTHROW message="CF_openLayer needs an ending tag! Please
ensure that this tag has an closing tag.">
</CFEXIT>
</cfif>

<!-- Set all default values for optional parameters -->
```

```
<CFPARAM NAME="ATTRIBUTES.border"
DEFAULT="Yes">
<CFPARAM NAME="ATTRIBUTES.bordersize"
DEFAULT="3">
<CFPARAM NAME="ATTRIBUTES.bordercolor"
DEFAULT="##F0F0F0">
<CFPARAM NAME="ATTRIBUTES.image"
DEFAULT="">
<CFPARAM NAME="ATTRIBUTES.imageopen"
DEFAULT=#ATTRIBUTES.image#>
<CFPARAM NAME="ATTRIBUTES.title"
DEFAULT="">
<CFPARAM NAME="ATTRIBUTES.titleclosed"
DEFAULT=#ATTRIBUTES.title#>
<CFPARAM NAME="ATTRIBUTES.titletooltip"
DEFAULT="">
<CFPARAM NAME="ATTRIBUTES.isclosed"
DEFAULT="No">
<CFPARAM NAME="ATTRIBUTES.width"
DEFAULT="100%">
<CFPARAM NAME="ATTRIBUTES.titlefont"
DEFAULT="Arial">
<CFPARAM NAME="ATTRIBUTES.titlecolor"
DEFAULT="">
<CFPARAM NAME="ATTRIBUTES.titlecoloropen"
DEFAULT=#ATTRIBUTES.titlecolor#>
<CFPARAM NAME="ATTRIBUTES.titlesize"
DEFAULT="10">

<!-- Make the title strings safe to use with JavaScript -->
<cfset ATTRIBUTES.title = Replace(ATTRIBUTES.title, '"',
"&acute;", "ALL")>
<cfset ATTRIBUTES.titleclosed = Replace(ATTRIBUTES.title-
closed, '"', "&acute;", "ALL")>

<!-- Store in a REQUEST-scoped variable if the accessing
browser is ok... -->
<cfif NOT IsDefined("REQUEST.openLayerBrowserSafeForIE4")>
    <!-- *** Browser-Check for Internet Explorer 4 and higher
*** -->
    <!-- Check for Internet Explorer -->
    <cfset tmpPositionMSIE =
ListContainsNoCase(CGI.HTTP_USER_AGENT, "MSIE",";")>
    <cfif tmpPositionMSIE GT 0>
        <!-- Check for Version 4.0 and higher -->
        <cfset tmpVersionMSIE =
ListLast(ListGetAt(CGI.HTTP_USER_AGENT,tmpPositionMSIE,";"),"
")>
        <cfif IsNumeric(ListFirst(tmpVersionMSIE,"."))>
            <cfif ListFirst(tmpVersionMSIE,".") GTE 4>
                <cfset REQUEST.openLayerBrowserSafeForIE4 = true>
            <cfelse>
                <cfset REQUEST.openLayerBrowserSafeForIE4 = false>
```

# **JDJ STORE**

**[www.jdjstore.com](http://www.jdjstore.com)**

```

</cfif>
<cfelse>
<cfset REQUEST.openLayerBrowserSafeForIE4 = false>
</cfif>
<cfelse>
<cfset REQUEST.openLayerBrowserSafeForIE4 = false>
</cfif> <!-- REQUEST.openLayerBrowserSafeForIE4 is "true"
when IEXP 4 and higher; otherwise "false" -->
</cfif>

<!-- ***** Here comes the actual work of this custom tag:
***** -->

<!-- When the starting-tag is being processed: -->
<CFIF ThisTag.ExecutionMode EQ "START">

    <!-- Generate a random unique ID for each instance of this
tag so that each layer can be accessed separately -->
    <cfset GeneratedUniqueLayerID = RandRange(1,999999)>

    <!-- Now set the starting properties depending on the
passed attributes... -->
    <cfif CompareNoCase(ATTRIBUTES.border, "Yes") IS 0>
    <cfset TMP.border = "solid">
    <cfelse>
    <cfset TMP.border = "none">
    </cfif>

    <cfif CompareNoCase(ATTRIBUTES.isclosed, "Yes") IS 0>
    <cfset TMP.display = "none">
    <cfset TMP.visibility = "hidden">
    <cfset TMP.borderwidth = "0">
    <cfset TMP.title = ATTRIBUTES.titleclosed>
    <cfif Len(ATTRIBUTES.image)>
    <cfset TMP.image = ATTRIBUTES.image>
    </cfif>
    <cfif Len(ATTRIBUTES.titlecolor)>
    <cfset TMP.titlecolor = ATTRIBUTES.titlecolor>
    </cfif>
    <cfelse>
    <cfset TMP.display = "block">
    <cfset TMP.visibility = "visible">
    <cfset TMP.borderwidth = ATTRIBUTES.bordersize>
    <cfset TMP.title = ATTRIBUTES.title>
    <cfif Len(ATTRIBUTES.image)>
    <cfset TMP.image = ATTRIBUTES.imageopen>
    </cfif>
    <cfif Len(ATTRIBUTES.titlecolor)>
    <cfset TMP.titlecolor = ATTRIBUTES.titlecoloropen>
    </cfif>
    </cfif>

<CFOUTPUT>

    <!-- Store in a REQUEST-scoped variable if this general
JavaScript-Code was already written on the page... -->
    <cfif NOT IsDefined("REQUEST.openLayerScriptAlreadyWrit-
ten")>

        <cfif REQUEST.openLayerBrowserSafeForIE4>
        <!-- The general JavaScript to expand and collapse the
layer... -->
        <SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
        <!--
        function switchLayer(lyrID, title, titleclosed, bor-
derwidth, imageA,imageB, colorA,colorB) {
            if (document.all) {
                // Determine if the layer is currently expanded or
collapsed:
                var tmpStatus;
                eval("tmpStatus = document.all.lyrGroup_" + lyrID
+ ".style.display");
                // OK, then now expand or collapse it...
                if (tmpStatus == 'none') {
                    eval("document.all.lyrGroup_" + lyrID + ".style.dis-

```

```

play='block';");
        eval("document.all.border_" + lyrID + ".style.border-
Width='"+borderwidth+"'");
        eval("document.all.title_" + lyrID
+ ".innerHTML='"+title+"'");
        tmpImage = imageA;
        tmpColor = colorA;
    }
    else {
        eval("document.all.lyrGroup_" + lyrID + ".style.dis-
play='none'");
        eval("document.all.border_" + lyrID + ".style.border-
Width='0'");
        eval("document.all.title_" + lyrID
+ ".innerHTML='"+titleclosed+"'");
        tmpImage = imageB;
        tmpColor = colorB;
    }
    // If an image is used as the click-control, switch
it...
    if ((tmpImage != null) && (imageA != imageB)) {
        eval("document.all.image_" + lyrID + ".src='"+tmpIm-
age+"'");
    }
    // If different title-colors are used, switch
them...
    if (tmpColor != '') {
        eval("document.all.title_" + lyrID
+ ".style.color='"+tmpColor+"'");
    }
}
//-->
</SCRIPT>
</cfif>

<cfset REQUEST.openLayerScriptAlreadyWritten = true>
</cfif>

<cfif REQUEST.openLayerBrowserSafeForIE4>
<!-- Draw the titled border -->
<fieldset id="border_#GeneratedUniqueLayerID#"
style="border-width:#TMP.borderwidth#; border-color:#ATTRI-
BUTES.BorderColor#; border-style:#TMP.border#; width:#ATTRI-
BUTES.width#">
    <legend>
    <span title="#ATTRIBUTES.TitleTooltip#">
    <!-- Draw either a button oder an image as click-
control... -->
    <cfif Len(ATTRIBUTES.image)>
    
    <cfelse>
    <input type="button" value=""
onClick="switchLayer(#GeneratedUniqueLayerID#,'#ATTRIBUTES.ti
tle#','#ATTRIBUTES.titleclosed#','#ATTRIBUTES.bordersize#,null
,null,'#ATTRIBUTES.titlecoloropen#','#ATTRIBUTES.titlecol-
or#')" style="width:12; height:12; margin-bottom:3">
    </cfif>
    <font face="#ATTRIBUTES.titlefont#">
    <b style="font-size:#ATTRIBUTES.titlesize#pt">
    <span id="title_#GeneratedUniqueLayerID#" <cfif
Len(ATTRIBUTES.titlecolor)>style="color:#TMP.titlecolor#"</cf
if>>TMP.title#</span>
    </b>
    </font>
    </span>
    </legend>
    <div id="lyrGroup_#GeneratedUniqueLayerID#" style="dis-
play:#TMP.display#; visibility:inherit; z-index:5; margin-

```



## Listing 2

<ATTRIBUTES>

```
<ATTRIB NAME="BORDER" TYPE="ENUMERATED">
  <ATTRIBOPTION CAPTION="Yes" VALUE="Yes"/>
  <ATTRIBOPTION CAPTION="No" VALUE="No"/>
```

**CODE LISTING**

The code listing for  
this article can also be located at  
[www.ColdFusionJournal.com](http://www.ColdFusionJournal.com)



[www.JavaDeveloper'sJournal.com](http://www.JavaDeveloper'sJournal.com)

# HotQuery from Array Software

REVIEWED BY  
**JERRY  
BRADENBAUGH**



**A**h yes, the almighty database. O highly exalted piece of technology, we worship you and the server rack upon which you are mounted.

Of course, once you have the data in there, how do you go about updating or even looking at it? What if you're a Web server host, and all your customers want remote access to their own data? It's pretty straightforward with tools such as Microsoft's Query Analyzer, Oracle's SQL \*Plus or Embarcadero's DBArtisan.

Of course, there's always a catch. The problem is, you need

the database client software on every machine you want to do this on. It's not likely that all your customers are going to have these tools. So why not do it through a Web browser – a freebie that anyone with a pulse has on their computer? How many of you out there thought, “Ya know, maybe I should write a Web-based app that could do that”? I know I did. Here's a good

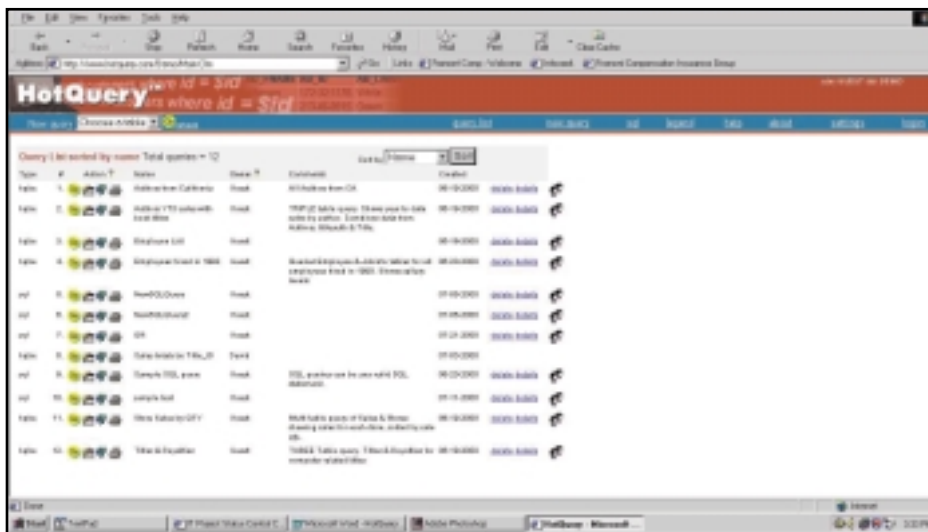


FIGURE 1: List of queries saved for the database

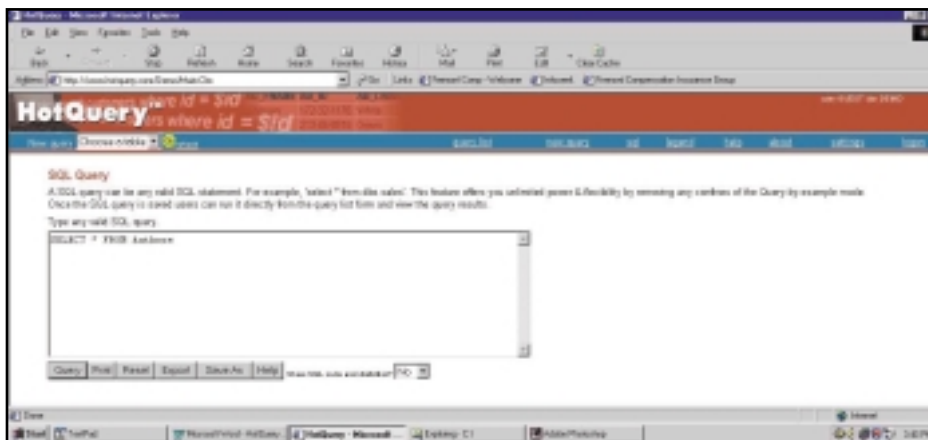


FIGURE 2: Modify your queries with straight SQL.

## VITALS

**Product Name:** HotQuery

**Vendor:** Array Software, [www.hotquery.com](http://www.hotquery.com)

**Functionality:** Web-based query tool

**Price:** \$1,950 (introductory offer)

### What's Hot

No database client software needed  
GUI- and text-based querying  
Export/e-mail features  
ColdFusion extensibility  
Great for Web server hosting

### What's Not

Installation can be a bit involved  
Lame documentation  
User interface not necessarily intuitive

one that made it past the “what if” stage. It's called HotQuery by Array Software, a ColdFusion solution to browser-based queries.

## The Basics

HotQuery is a browser-based query tool that supports SQLServer and Oracle databases. You simply log in by providing your HotQuery username and password (which is actually a user on the database), the ODBC datasource name and the type of database (SQLServer or Oracle). Once you're in, HotQuery displays a list of any queries already saved for this database, shown in Figure 1. From there you can run, edit, preview your existing queries and create new queries at will.

You can modify your queries in two ways. First, you can use straight SQL, as you would in any other query tool (see Figure 2). Just enter the columns, tables and joins you want to see, then preview or save your work.

The other way to modify queries is more of a GUI approach called

# Career Opportunities

# Career Opportunities





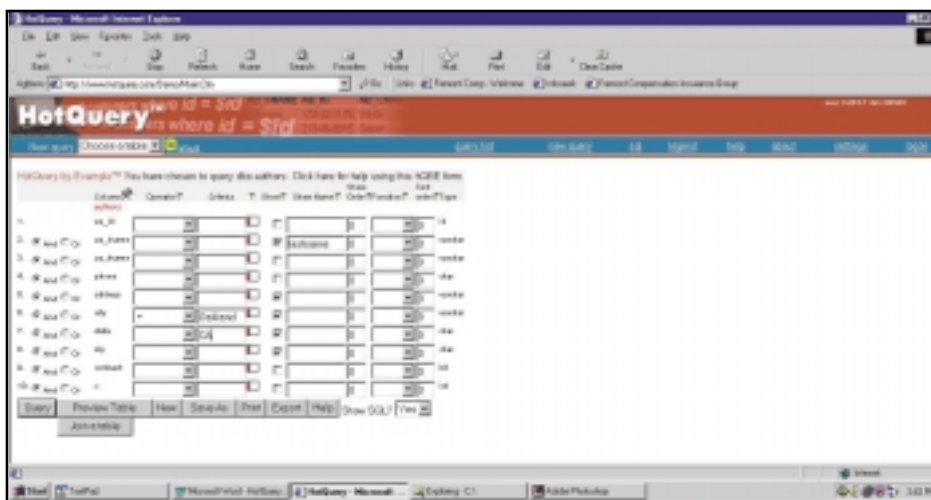


FIGURE 3: HotQuery-By-Example

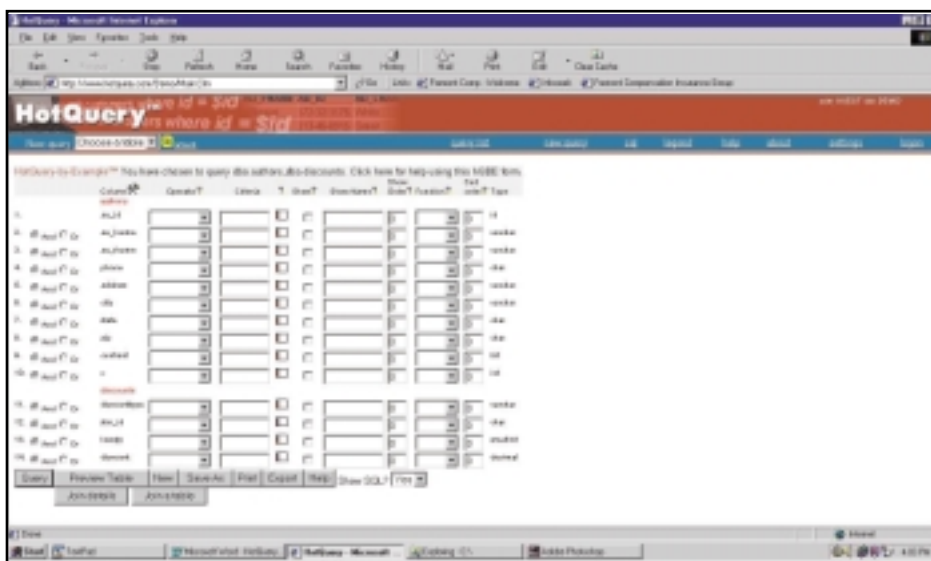


FIGURE 4: Confirmed join reflecting two tables

HotQuery-By-Example. Each row on the page represents a column in the table you choose. You choose the columns you want to see based on criteria. Here's how it works. Suppose I want to view the results from the following SQL query:

```
SELECT au_lname AS lastname, address,
city, state, zip FROM Authors WHERE
city = 'Oakland' and state = 'CA'
```

Figure 3 shows the HotQuery-By-Example equivalent. Note that each column to be displayed is checked in the respective Show? checkbox; the ShowName? field represents the AS co-relator, and the Operator? and Criteria fields on the page represent the WHERE clause syntax. It takes some getting used to, but before long you'll have it down.

Of course, it won't be long before you'll want to start joining tables. On the same HotQuery-By-Example page, you can choose Join a table. This displays a list of available tables for you to join (except for the table you were currently working with). Just choose a table, and this loads a list of available columns from your table of choice. The next step is to join this column with a column in your original table. When you've confirmed the join, the HotQuery-By-Example form will now reflect two tables. If you want to join more, you go again.

HotQuery lets you save your finished queries on the server and gives any user access to them. You can also export the output of those queries to text files. You choose the filename and delimiter. Add to that

the option to e-mail your query to anyone on the planet.

## Administration

HotQuery also allows you to adjust the settings of the user experience via its administration interface. The interface includes settings adjustments such as default data-source specification, admin username/password edition, system table display control, Java grid usage and record count display (per page).

## One Step Farther

If you like the basic functionality, but want to change things more to your liking, HotQuery comes with 100-plus custom ColdFusion tags. For example, you can use the custom tags to change which columns are displayed in query results based on the user.

## Different Flavors

HotQuery comes in three licensed versions: Corporate, ISP and Development. They all cost the same (\$1,950), and differences among them are slight. The Corporate version allows you to deploy HotQuery on any server in your corporation, and you get direct tech support from Array Software. The ISP license allows you to deploy it on any box on your server host domain, but Array Software provides only indirect tech support (HotQuery users would first look to the Web server host for support – Array would be second tier). The Development version is essentially the same as the Corporate license, but includes documentation (not available at the time of this review) for implementing the custom ColdFusion extensibility. Direct support is available.

## Wrapping Up

HotQuery is a good browser-based query tool that can save you the burden of relying on a database client. Though the help documentation wasn't available to me, the interface is relatively easy to use and the learning curve isn't that steep. Make sure you check out the demo at [www.hotquery.com](http://www.hotquery.com).



## ABOUT THE AUTHOR

Jerry Bradenbaugh is an independent Web consultant in Los Angeles. He's the author of *The JavaScript Application Cookbook* (O'Reilly) and maintains HotSys – the JavaScript resource ([www.serve.com/hotsys](http://www.serve.com/hotsys)).

# **CTIA**

**[www.ctia.com](http://www.ctia.com)**



## Developing Wireless Applications with ColdFusion

Facing the Challenges Part 2

BY  
CHARLES  
AREHART

## Some problems and solutions of venturing into wireless programming with CF

It's fun developing wireless applications in ColdFusion, but if you don't solve several key challenges it'll be more painful than pleasant.

In this second part of a series, I'll focus on some common problems encountered by CF/WML developers.

In the last issue the focus was on getting you started: an introduction to WAP (the Wireless Access Protocol), the early state of wireless programming, how to get a simulator to begin testing code, resources for learning more about WAP and the markup language for WAP phones, WML (Wireless Markup Language).

I intentionally didn't focus on WML itself or on the details of the WAP architecture. These matters are best left to the many SDK documents, Web sites and the growing number of books on the subject.

In this issue I'll focus on matters not covered by those resources: the problems you'll typically encounter when getting started with wireless programming in conjunction with ColdFusion. There are indeed many challenges that would affect any WML developer. I'll touch on a couple of them, but the real focus is on the CF-specific challenges, as well as some tips and tricks for getting things to work when trouble strikes.

### The Key to Doing WML in CF

Before discussing the common challenges, I ought to at least give you a bit of sample code to begin with in case this is your first time reading about WML. As mentioned in the last article, I don't want to repeat too much of what was covered by Ben Forta's intro to WML in his December 1999 article, "No Strings Attached" (Vol. 1, issue 6). Please see that for more introductory WML.

The bottom line is that all you need is a properly formatted CFCCONTENT tag to indicate to the

browser that the code you're building is WML rather than HTML (ColdFusion's default). This sets the MIME type of the page you're creating. Simply include this at the top of your program:

```
<CFCCONTENT TYPE="text/vnd.wap.wml">
```

That's not quite all you need. There's a set of basic "prologue" statements that must be specified at the top of any WML page, so the more complete definition of the beginning of a CF/WML page, as offered in a simple "Hello World" example, is:

```
<CFCCONTENT
TYPE="text/vnd.wap.wml"><?xml ver-
sion="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD
WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.
xml">
<wml>
<card>
<p>
Hello World
</p>
</card>
</wml>
```

Notice that we've put the first two lines of code on the same line. Many WML developers have found that the ?xml declaration line should be coded without any carriage returns preceding it.

From here you can use any CF code to generate any WML as needed by your application. Again, see other more general-interest WML resources for more about the WML you can use. And you can do this in any version of CF. This leads to a natural question often asked by new CF/WML devel-

opers: What version of ColdFusion allows WML programming?

When people notice that Allaire has highlighted new WML support in 4.5, they get the impression that WML can only be done in 4.5 of CF Server. In fact, the new support is really just extensions to CF Studio and HomeSite to support creating, editing and getting help on WML tags, and a new page wizard. We'll review those in a later article.

Any version of ColdFusion (Server and Studio) can support WML programming if you're coding the tags by hand, since all you need is that CFCCONTENT tag at the top of the page.

### The Inevitable Errors You'll Encounter

If you try this sample code, or when you begin doing more substantial code, you'll inevitably encounter errors. The challenge is determining if it's a CF error or a WML error, and even then what works in one phone may fail in another due to the incompatibilities of phones supporting WML in different ways (which is beyond the scope of this article). But there are some simple issues we can deal with.

### Viewing the Code in the Right Browser

First of all, you can't expect to run the preceding code sample from within a browser like IE or Netscape (at least not the current versions). With the CFCCONTENT tag in place, you can view the output of the page only in a browser that supports WML (phones or phone simulators, as discussed in the last article). If you try to view it in a normal browser,

# **SD 2000 EAST**

**[www.sdexpo.com](http://www.sdexpo.com)**

you'll likely receive a prompt to save the file being sent, since the browser doesn't know what to do with a file of the MIME type specified in the CFCONTENT tag.

### Dealing with "Invalid Content" or "Syntax Errors"

Even when viewed on a real phone (or simulator), you'll encounter errors in your first forays into WML. There are a few things to keep in mind.

First, WML is case-sensitive. You must use lowercase for WML tags and attributes. Also, WML follows the rules of XML programming, so all tags must have a closing tag. Even a `<p>` tag must have a closing `</p>`, unlike in HTML, and tags like `<br>`, which have no closing tag, must be written as `<br/>`.

Note that you can't send HTML tags to a WML browser. (Well, that's not entirely true. Some WAP gateways can convert HTML to WML, but that's not universal and shouldn't be relied on.) If you send HTML to WML browsers, they'll generally "choke" on it. So be careful – don't intermix HTML and WML. You really need to find some WML resources to learn more. For all their similarity, there are many significant differences between HTML and WML programming, and they're more than just differences in tag names.

Other problems could arise if you have an error in your CF code (causing CF to display an error message), or if you have server-side debugging information turned on, both of which generate HTML, as we'll discuss next.

### Turn Off Server-Side Debugging

ColdFusion's HTML heritage can really get in the way when doing WML programming. When debugging is turned on in the Administrator, you normally see the debugging info at the bottom of any CF page. But that information is sent in HTML formatting to the browser, which is fine in a normal browser, but it's death to most WML browsers since they can't read HTML.

You don't need to turn off debugging in the Administrator. As of ColdFusion release 4, you have the option to turn off debugging with the following line of code:

```
<CFSETTING SHOWDEBUGOUTPUT="no">
```

This will disable debugging until you turn it on again or until the currently running template ends. If you're running a release of ColdFusion prior to release 4, it's probably best to just turn off server-side debugging.

### Why Can't I See ColdFusion Error Messages?

On a related matter, the same problem of CF's HTML heritage has to do with error message handling. When you get an error in CF, you normally see the CF error message display indicating the error and the page and line number of the code in error. But that error is also sent in HTML formatting to the browser, which chokes the WML browser.

This would seem a real dilemma. You can't see the error because the phone doesn't recognize it as valid WML. So how are you supposed to deal with it? Well, fortunately, Allaire has put in place error-handling components to improve the ability of the developer to control the error message display to the user.

You might think that CFERROR, and its ability to create a specially formatted error page, might come to the rescue. Unfortunately, the two older forms of CFERROR pages (type="request" and type="validation") send pure HTML (and an HTML MIME type), which you can't override. Since you can't put any CF tags on those error pages, you can't even add a CFCONTENT.

(This also means, of course, that you can't use CF's hidden field validation to do server-side form validation. The validation error page that it offers is also pure HTML and you can't override it.)

Perhaps Allaire will offer a solution to these dilemmas in the future. For now, if you have release 4.5, you can in many cases at least solve some of the first problem – seeing CF error messages – by taking advantage of the new CFERROR type="exception". This transfers control when an error occurs to a page that is allowed to do CF tags (an incredibly valuable new feature in and of itself), including a CFCONTENT tag. You can then format the error for display to the WML browser.

### Formatting the Error Message Itself

You're not quite done, though. The CF error message itself, offered inside such a CFERROR page as the variable CFERROR.Diagnostics, will usually be HTML-formatted. This was fine when sending it for display to an HTML browser, but again it's a problem in WML. Fortunately, you can solve it by using a CF function to convert the HTML tags in the message into their corresponding display codes, such as `&gt;` for `>` and `&lt;` for `<`.

Allaire offers two solutions for this. In 4.5 there's an XmlFormat() function for just this purpose. In older releases you can use HtmlEditFormat(). It's not the perfect solution, as it'll simply cause any HTML tags in the error message to be displayed to the screen, but it's easier than trying to strip out the HTML tags.

This approach can also be used to format error messages displayed in a CFCATCH (using the "cfcatch.detail" variable).

You may want to create an application.cfm that sets the CFSETTING to disable debugging output (on a release 4 or above server) and sets a CFERROR type="exception" (on a release 4.5 or above server) to point to a template that can format the error message in WML.

Keep in mind that none of this CF error handling will help if the error in your page is due to badly formed WML. In this case you'd do well to have a simulator that allows you to see the actual WML being sent to the browser (most real phones won't provide access to that information).

### Form Processing Works in Simulator, Fails in Real Phone

This is another thorny problem, and it's not generally CF-specific. Some background will help. When you test code in most simulators, you may be using a form of communication between the simulator and the Web server that's often called "http direct". There is direct communication between the simulator and the server.

This is fine, but real phones use a form of communication that goes from the phone to a gateway (usually within the phone company provider) that then communicates to the Web server on the phone user's behalf. (It acts as both a proxy

and a translator, since what's really sent to real phones is an encoded form of WML to minimize data traffic.)

The problem with this scenario is that code that works fine in the "http direct" mode may fail when run on a real phone or in the simulator's gateway mode.

A perfect example of this is "form processing" by way of the "post" method. If the example in Listing 1 is submitted on a simulator, it will work, but on a real phone it may fail.

I don't have room here to explain the unique differences of form processing in WML versus HTML, but the example shows many of the significant features. Note that there's no "form" tag per se, and that the form data is gathered in one place (the input tag), stored in a variable on the phone ("stock") and passed to the server in yet another tag (<go>) when the user presses the "accept" button or equivalent. Very different from HTML! But note the powerful new formatting possibilities in the <input> tag. This allows only up to four alphanumeric characters.

The problem occurs when the page itself is submitted with method="post". In testing on a simulator in HTTP direct mode, this code works fine. In a real phone, however, which goes through a gateway, it will fail to submit. In the Phone.com simulator, using the "up.link" mode to go through a gateway, it fails with an error "500:internal server error". It seems the gateway is simply unable to make a "post" method call to the server.

And it doesn't seem to be restricted to CF Server (I tested the problem against an ASP program and got the same error). It may be restricted to Phone.com gateways, though I've heard it happened to users of Nokia phones and simulators.

The short-term solution is to change the method to "get". ColdFusion will receive the variables as "url." variables (in the example above, the "url.symbol"). Unlike in HTML browsers, this doesn't have the same problem of showing the user the form variables on the URL of the action page since phone users don't see the URLs of sites they're visiting (currently).

There is the security risk that on an SSL-enabled connection (called WTLS in the WAP world) the data passed on the URL isn't encrypted, so this isn't a solution that will work when passing userids and passwords, or credit cards if the goal is to encrypt that data. But that, too, is beyond the scope of this article.

### That's All for Now

That's all the time (or rather, room) we have this month. Upcoming articles in this series will cover some other challenges (and solutions) regarding wireless development, such as doing browser detection to serve both HTML and WML browsers, handling sessions in an environment in which cookies aren't always supported, doing CFLOCATION (you have to use Web server root-relative paths, even to send to a file in the same directory), what to do when CFCONTENT simply doesn't work in your CF server, processing multivalued form field submissions, sending notifications and pushing messages to the phone (using COM objects), and the support in Studio for WML.

You can learn about WML and generic challenges as well as more about these CF challenges in the recently released book *Professional WAP* (Wrox Press). I have a chapter in the book on CF/WML programming. Other WML/WAP books will soon be in the market as well.



### ABOUT THE AUTHOR

*Charles Arehart is a certified Allaire trainer and CTO of SysManage, an Allaire partner. He contributes to several CF resources and is a frequent speaker at user groups throughout the country.*

# INTELLIANT

[www.inteliant.com](http://www.inteliant.com)

CAREHART@SYSTEMANAGE.COM

```
<?CONTENT TYPE="text/vnd.wap.wml"><?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card>
  <do type="accept">
    <go href="wml_action.cfm" method="post">
      <postfield name="symbol" value="\$(stock)"/>
    </go>
  </do>
<p>
  Enter stock symbol:
  <input name="stock" format="4A"/>
</p>
</card>
</wml>
```

ADVERTISER	URL	PH	PG
ABLECOMMERCE	WWW.ABLECOMMERCE.COM	360.253.4142	4
ABLECOMMERCE	WWW.AUCTIONBUILDER.COM	360.253.4142	2
ALLAIRE	WWW.VUE.COM/ALLAIRE	877.460.8679	37
ALLAIRE	WWW.ALLAIRE.COM	888.939.2545	41
ALLAIRE DEVELOPER CONFERENCE	WWW.ALLAIRE.COM/CONFERENCE		29
CAREER OPPORTUNITIES		800.582.3089	57
CFXHOSTING	WWW.CFXHOSTING.COM	800.939.8188	17
COMPUTERWORK.COM	WWW.COMPUTERWORK.COM		66
CORDA TECHNOLOGIES	WWW.POPCHART.COM	888.763.0517	11
CTIA	WWW.CTIA.NET		59
DESIGNING SOFTWARE	WWW.DESIGNINGSOFT.COM		66
DEVELOPERSNETWORK	WWW.DEVELOPERSNETWORK.COM	416.203.3690	39
DIGITALNATION	WWW.DEDICATEDSERVER.COM	877.624.7897	15
EVOLUTION B	WWW.EVOLUTION.COM/SYNERGYFORFREE		67
FALL INTERNET WOTLD	WWW.PENTONEVENTS.COM		47
FIG LEAF SOFTWARE	WWW.FIGLEAF.COM	202-797.5478	3
INTELIANT	WWW.INTELIANT.COM	800.815.5541	63
INTELLIGENT ENVIRONMENTS	WWW.SCREENSURFER.COM		21
INTERLAND	WWW.INTERLAND.COM	800.815.5541	9
INTERMEDIA	WWW.INTERMEDIA.NET	800.379.7729	68
JAVA DEVELOPER'S JOURNAL	WWW.JAVADEVELOPERSJOURNAL.COM	201.802.3021	55
JAVACON2000	WWW.JAVACON2000.COM		49
JDJ STORE	WWW.JDJSTORE.COM	888.303.JAVA	53
MACROMEDIA	WWW.MACROMEDIA.COM	415.252.2000	12,13
MISTRAL DESIGN GROUP	WWW.PURPLEHOSTING.COM		35
NETDIVE	WWW.NETDIVE.COM		51
SD2000	WWW.SDEXPO.COM	800.441.8826	61
SHIFT 4	WWW.SHIFT4.COM	800.265.5795	25
SITEHOSTING.NET	WWW.SITEHOSTING.NET	877.NTHOSTING	33
SYS-CON MEDIA INC.	WWW.SYS-CON.COM	800.513.7111	57
THE SHORT LIST	WWW.THESHORTLIST.COM		23
VIRTUALSCAPE	WWW.VIRTUALSCAPE.COM	212.460.8406	19,27
WIRELESS DEVCON	WWW.WIRELESSDEVCON2000.COM		44,45
WOODBOURNE SOLUTIONS	WWW.WOODBOURNESOLUTIONS.COM	301.428.7620	43
XML DEVCON 2000-2001	WWW.XMLDEVCON2000.COM	800.513.7111	31

3007 Kingston Road Toronto, Ontario CANADA M1M 1P1  
www.developersnetwork.com • 416-203-3610



## digitalNATION - a VERIO company

digitalNATION, VERIO's Advanced Hosting Division, is the world's leading provider of dedicated server hosting, with over 1,650 servers online today. dN's superior connected network and service abilities have earned dN a solid reputation as a first-choice provider of dedicated server solutions (Sun, Windows NT, Linux and Cobalt). digitalNATION has been providing online and network services for over six years. One of the first ISPs to provide dedicated servers running Microsoft Windows NT, the dN staff has unparalleled experience in this industry.

5515 Cherokee Ave, Alexandria, VA 22312-2309

[www.dedicatedserver.com](http://www.dedicatedserver.com) • 703 642-2800

## FigLeaf Software

Fig Leaf Software specializes in developing turnkey web database applications and dynamic, data-driven websites. Our goal is to develop web-based client-server applications with functionality and interface design that are nearly indistinguishable from desktop software developed using traditional tools such as Visual Basic, Visual FoxPro, Delphi, or C. Above all, we want to bring maximum value to our clients at the minimum cost. The key to fulfilling this is ensuring our staff members are experts in their particular field. Our clients expect excellence, and we demand it of ourselves.

1400 16th St. NW, Suite 220, Washington, DC 20036

[www.figleaf.com](http://www.figleaf.com) • 877.344.5323

## Inteliant

Inteliant Corporation, a leading ColdFusion consulting firm, has an outstanding reputation for providing highly skilled developers for Internet, Intranet, Extranet, Software Development, or any ColdFusion application. Our national practice has emerged to meet the evolving needs of our clients by providing resources onsite or developing remotely. Our company provides the most cost effective service in the industry and we strive to add value to your projects by minimizing expenses whenever possible. Inteliant... "Delivering Intelligent Solutions

1150 Hancock Street, Suite 4, Quincy, MA 02169

[www.inteliant.com](http://www.inteliant.com) • 800-815-5541

## Intelligent Environments

iE enables financial services and insurance companies to offer comprehensive, integrated, full-service web sites quickly and cost effectively. We are able to do this because of the unique combination of our off-the-shelf net finance applications and our integration technology.

500 Unicorn Park Drive, Suite 301, Woburn, MA 01801

[www.ieinc.com](http://www.ieinc.com) • 781-932-1700

## Interland

Interland, Inc., ranked the No. 1 Web Hosting Provider for small- to medium-sized businesses by Windows NT Magazine, Network Computing and PC Magazine, is a global leader in Web hosting solutions ranging from a basic Web site to a sophisticated e-commerce storefront. Interland proudly features 24-hour, 7-day toll-free technical support and an advanced Administration Page. By deploying the best products, services, and support in the industry, Interland can build a Web presence that works for you. Speed. Reliability. Support. - Guaranteed.

101 Marietta Street, Second Floor, Atlanta, GA 30303

[www.interland.com](http://www.interland.com) • 800-214-1460

## Intermedia, Inc.

Our advanced virtual hosting packages (powered by Microsoft Windows NT and Internet Information Server 4.0) offer an environment supporting everything today's advanced Web developer or sophisticated client could ask for. Complete ODBC support is available on plans B and C. We support Microsoft Index Server on all hosting plans.

953 Industrial Avenue, Suite 121, Palo Alto, CA 94303

[www.intermedia.net](http://www.intermedia.net) • 650 424-9935

## Macromedia

New Macromedia UltraDev lets you create database-driven Web applications faster than ever before. It also allows you to create ASP, JavaServer Pages, and CFML applications in a single design environment. So whether you love working directly with source code, or prefer to work visually, cut the time it takes to create employee directories, product catalogs, database search pages and more.

600 Townsend Street, San Francisco, CA 94103

[www.macromedia.com](http://www.macromedia.com) • 415 252-2000

## Sitehosting.NET

Successful electronic commerce starts at SiteHosting.net; a division of Dynatek Infoworld, Inc., which provides total Web development services. We offer personal and efficient customer service with reliability at value prices. All our plans include access to SSL (Secure Socket Layer). We support ColdFusion, Active Server Pages, Real Audio/Video, Netshow Server, and more. Our hosting price starts at \$14.95/month.

13200 Crossroads Parkway North, Suite 360,

City of Industry, CA 91746

[www.sitehosting.net](http://www.sitehosting.net) • 877 684-6784

## The Short List

Short on ideas? THEshortLIST.com offers a premier online, database-driven, résumé databank for IT and e-commerce professionals. We allow leading employers and top IT and e-commerce professionals to find each other online in minutes. Our unique system matches employers with qualified candidates. No more fruitless interviews.

3181 Allison Drive, Gibsonia, PA 15044

[www.theshortlist.com](http://www.theshortlist.com) • 877 482-8827

## Woodbourne Solutions

We take your web project idea and make it a reality, on-budget and on-time, guaranteed. We offer complete solutions analyzing your business needs, creating your complete site (application, database, and user interface), testing it out, hosting it, and making sure it's secure.

Intelligent Environments

12401 Middlebrook Road, Suite 160, Germantown, MD 20874

301-428-7620

## Virtualscape

Why host with Virtualscape? Nobody else on the Internet understands what it takes to host ColdFusion like we do. Virtualscape is the leader in advanced Web site hosting. From Fortune 500 extranets to e-commerce sites and more, developers recognize our speed, stability, reliability and technical support.

215 Park Avenue South, Suite 1905, New York, NY 10003

[www.virtualscape.com](http://www.virtualscape.com) • 212 460-8406

To place an  
ad in the  
**ColdFusion  
Marketplace**  
contact Robyn Forma  
at 201 802-3022

**Express Technologies Launches CFXHosting.com**  
(Louisville, KY) – Express Technologies, Inc., introduces CFXHosting.com, a complete Web hosting service designed for programmers using Allaire's ColdFusion development language.

CFXHosting.com is the first company to provide a complete turnkey development solution for ColdFusion professionals. For \$99.95 per



month customers can host multiple sites and have complete access to features such as CFFILE, CFCONTENT, CFDIRECTORY, CFX tags and the ColdFusion Administrator. CFXHosting.com also provides full support for Microsoft SQL Server 7.0.

[www.cfxhosting.com](http://www.cfxhosting.com)

## Dot Hill Delivers State-of-the-Art Software Tools for Dynamic Volume Management

(Carlsbad, CA) – Dot Hill Systems Corp. announces enhancements to its suite of storage area



network (SAN) management software applications SANpath 3.1 and SANscape 2.3. SANpath 3.1 now delivers dynamic LUN

assignment, an advanced time-saving capability that allocates and reallocates server storage resources on the fly without interrupting critical applications. SANpath 3.1 also adds Linux, HP-UX and Windows 2000 to its list of supported platforms. SANscape 2.3 is now even more tightly integrated with SANpath and provides IT managers with an intuitive, easy to-use drag-and-drop user

interface for managing storage allocation operations, and also adds HP-UX, Linux, Windows 2000 and Netware to its list of supported platforms. Both applications work with Dot Hill's carrier-class SANnet storage solutions to enhance performance, simplify storage management and protect against component failures.

[www.dothill.com](http://www.dothill.com)

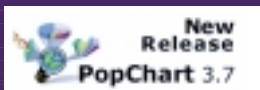


## CORDA Releases PopChart 3.7

((Orem, UT) – CORDA Technologies, Inc., introduces

PopChart 3.7, a powerful collection of tools available to create and display dynamic charts and graphs for Internet and intranet sites.

Along with enhancements to the installation process and increased ease of use,



PopChart 3.7 Image Server Pro includes new support for

image maps, allowing advanced drill-down capabilities with GIF

and PNG images, more graph types to choose and create, improved caching and an enhanced help system.

[www.corda.com](http://www.corda.com)

## CFDev.com.com Launches Activescan

(Utica, NY) – CFDev.com announces the release of Activescan, a development tool that allows users to send images

directly from their scanner to the Web server, making it easier for clients to publish pictures of property, their portfolio, products or anything else that can be scanned.

[www.cfdev.com.com/activescan/executive.cfm](http://www.cfdev.com.com/activescan/executive.cfm)



## COLD FUSION Developer's Journal

### Look What's Coming!

Attention Advertisers:

Don't Miss your chance to advertise in the Allaire Developer Conference

**Special Issue!** Call Robyn Forma  
201-802-3022

or email [robyn@sys-con.com](mailto:robyn@sys-con.com)



# Designing Software

[www.designingsoft.com](http://www.designingsoft.com)

# Computerwork.com

[www.computerwork.com](http://www.computerwork.com)

# **Evolution B**

**[www.evolution.com/synergyforfree](http://www.evolution.com/synergyforfree)**

# **Intermedia**

**[www.intermedia.net](http://www.intermedia.net)**